



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  
НАПРАВЛЕНИЕ ПОДГОТОВКИ **09.04.01 Информатика и вычислительная техника**  
МАГИСТЕРСКАЯ ПРОГРАММА **09.04.01/07 Интеллектуальные системы анализа,  
обработки и интерпретации больших данных.**

## Отчет

по лабораторной работе №5

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

Студент

ИУ6-23М

(Группа)

В.А Антонов

(Подпись, дата)

(И.О. Фамилия)

Преподаватель

П.В. Степанов

(Подпись, дата)

(И.О. Фамилия)

Москва, 2022

## Лабораторная работа №5

### Задание: Вариант 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.

### Ход работы: Код программы файла VectorR3

```
import java.util.ArrayList;

public class VectorR3 {

    private double x1;
    private double x2;
    private double y1;
    private double y2;
    private double z1;
    private double z2;

    private double x;
    private double y;
    private double z;

    public VectorR3() {
    }

    public VectorR3(double x1, double x2, double y1, double y2, double z1,
double z2) {

        this.x1 = x1;
        this.x2 = x2;
        this.y1 = y1;
        this.y2 = y2;
        this.z1 = z1;
        this.z2 = z2;

        this.x = x2 - x1;
        this.y = y2 - x1;
        this.z = z2 - z1;

    }

    public double getX() {
        return x;
    }

    public double getY() {
        return y;
    }

    public double getZ() {
        return z;
    }
}
```

```

@Override
public String toString() {
    return "VectorR3: " +
        "x1 = " + x1 +
        ", x2 = " + x2 +
        ", y1 = " + y1 +
        ", y2 = " + y2 +
        ", z1 = " + z1 +
        ", z2 = " + z2+
        '.';
}
}

```

### Код программы файла MainVectorR3

```

import java.util.*;
public class MainVectorR3 {
    public static void isCoplanar(VectorR3 v1, VectorR3 v2, VectorR3 v3) {
        double m = v1.getX() * v2.getY() * v3.getZ() + v1.getY() * v2.getZ() *
        * v3.getX()
            + v1.getZ() * v2.getX() * v3.getY() - v1.getZ() * v2.getY() *
        v3.getX()
            - v1.getX() * v2.getZ() * v3.getY() - v1.getY() * v2.getX() *
        v3.getZ();
        if (m == 0) {
            System.out.println("Вектора компланарны");
        } else {
            System.out.println("Вектора не компланарны");
        }
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        int number = 1;
        System.out.print("Введите количество векторов: ");
        try {
            number = scanner.nextInt();
        }

        catch (Exception e){
            System.out.println("Ошибка количество векторов");
        }

        VectorR3[] vector_array = new VectorR3[number];
        Random random = new Random();
        double x1, x2, y1, y2, z1, z2;
        System.out.println("Введите координаты векторов: ");
        for (int i = 0; i < number; i++) {
            x1 = random.nextDouble();
            x2 = random.nextFloat();
            y1 = random.nextFloat();
            y2 = random.nextFloat();
            z1 = random.nextFloat();
            z2 = random.nextFloat();
            vector_array[i] = new VectorR3(x1, x2, y1, y2, z1, z2);
        }

        for (int i = 0; i < number; i++) {

```

```

        System.out.println("Вектор " + i + ": " +
vector_array[i].toString());
    }

    for (int i = 0; i < number - 2; i++) {
        for (int j = i + 1; j < number - 1; j++) {
            for (int k = j + 1; k < number; k++) {
                System.out.print("Вектора " + i + ", " + j + ", " + k +
": ");

                isCoplanar(vector_array[i], vector_array[j],
vector_array[k]);
            }
        }
    }
}
}
}

```

## Код программы файла MatrixHandler

```

import java.util.Arrays;

public class MatrixHandler {

    public static int[][] add (int[][] a, int[][]b) throws
IllegalArgumentException{
        if (a.length == b.length && a[0].length == b[0].length){
            int[][] c = new int[a.length][a[0].length];
            for (int i = 0; i < a.length; i++){
                for (int j = 0; j < a[0].length; j++){
                    c[i][j] = a[i][j] + b[i][j];
                }
            }

            return c;
        }

        else throw new IllegalArgumentException ("Размеры матрицы должны быть
одинаковыми");
    }

    public static int[][] subtract (int[][] a, int[][]b) throws
IllegalArgumentException{

        if (a.length == b.length && a[0].length == b[0].length){
            int[][] c = new int[a.length][a[0].length];
            for (int i = 0; i < a.length; i++){
                for (int j = 0; j < a[0].length; j++){
                    c[i][j] = a[i][j] - b[i][j];
                }
            }

            return c;
        }

        else throw new IllegalArgumentException ("Размеры матрицы должны быть
одинаковыми");
    }
}

```

```

        public static int[][] multiply (int[][] a, int[][]b) throws
        IllegalArgumentException{
            int l1 = a.length;//m
            int l2 = b[0].length;//n
            int l3 = b.length;//o
            if (a.length == b[0].length){
                int[][] c = new int[a.length][b[0].length];
                for (int i = 0; i < l1; i++) {
                    for (int j = 0; j < l2; j++) {
                        for (int k = 0; k < l3; k++) {
                            c[i][j] += a[i][k] * b[k][j];
                        }
                    }
                }
                return c;
            }
            else throw new IllegalArgumentException ("Количество столбцов матрицы
A должно совпадать с количеством строк матрицы B.");
        }

        public static int[] sumRows(int [][]a)
        {
            int []array = new int[a.length];
            for(int i = 0;i<a.length;i++)
            {
                for(int j = 0;j<a[0].length;j++)
                {
                    array[i]+=Math.abs(a[i][j]);
                }
            }
            System.out.println(Arrays.toString(array));
            return array;
        }

        public static int[] sumCols(int [][]a)
        {
            int[]array = new int[a[0].length];
            for(int j = 0;j<a[0].length;j++)
            {
                for(int i =0;i<a.length;i++)
                {
                    array[j]+=Math.abs(a[i][j]);
                }
            }
            System.out.println(Arrays.toString(array));
            return array;
        }

        public static int maxRows(int []a)
        {
            int max = 0;
            for(int val:a) {
                if(val > max)max = val;
            }
            return max;
        }

        public static int maxCols(int []a)
        {
            int min = a[0];
            for(int val:a) {

```

```

        if(val > min) min=val;
    }
    return min;
}

//вывод матрицы на экран
public static void print_add(int [][] c){
    System.out.println("Сложение: ");
    for (int i = 0; i < c.length; i++){
        for (int j = 0; j < c[0].length; j++){
            System.out.print(c[i][j] + " ");
        }
        System.out.print("\n");
    }
}

public static void print_subtract(int [][] c){
    System.out.println("Вычитание: ");
    for (int i = 0; i < c.length; i++){
        for (int j = 0; j < c[0].length; j++){
            System.out.print(c[i][j] + " ");
        }
        System.out.print("\n");
    }
}

public static void print_multiply(int [][] c){
    System.out.println("Умножение: ");
    for (int i = 0; i < c.length; i++) {
        for (int j = 0; j < c[0].length; j++) {
            System.out.format(c[i][j] + " ");
        }
        System.out.print("\n");
    }
}

public static void Norm1(int [][] a){
    System.out.println("Norm1 =" + maxRows(sumRows(a)));
}

public static void Norm2(int [][] a){
    System.out.println("Norm2 =" + maxCols(sumCols(a)));
}

}

```

## Код программы файла Matrix

```

import java.util.Scanner;

public class Matrix {
    public static void main(String[] args) {
        System.out.print("Введите диапазон n: ");
        Scanner scan = new Scanner(System.in);
        int n = 1;
        try{
            n= Integer.parseInt(scan.nextLine());
        }
        catch (Exception e){
            System.out.println("Ошибка ввода диапазона");
        }

        int[][] a = new int[n][n];
    }
}

```

```

int[][] b = new int[n][n];

for (int i = 0; i < n; i++) {
    for (int j = 0; j < n; j++) {
        a[i][j] = (int) (Math.random() * n );
        b[i][j] = (int) (Math.random() * n );
    }
}

System.out.println("Матрица A: ");
for (int i = 0; i < n; i++) { //вывод матрицы
    for (int j = 0; j < n; j++) {
        System.out.print(a[i][j] + " ");
    }
    System.out.print("\n");//переход на новую строку
}

System.out.println("Матрица B: ");
for (int i = 0; i < n; i++) { //вывод матрицы
    for (int j = 0; j < n; j++) {
        System.out.print(b[i][j] + " ");
    }
    System.out.print("\n");//переход на новую строку
}
int [][] c = new int[n][n];
MatrixHandler.print_add(MatrixHandler.add(a, b));
MatrixHandler.print_substract(MatrixHandler.substract(a, b));
MatrixHandler.print_multiply(MatrixHandler.multiply(a, b));
MatrixHandler.Norm1(a);
MatrixHandler.Norm2(a);
// теперь проверим выбрасывание исключений
try {
    MatrixHandler.add(c, b);
}
catch (IllegalArgumentException e) {
    System.out.println("Неправильные аргументы, введенные в метод
добавления");
}

try {
    MatrixHandler.multiply(a, b);
}
catch (IllegalArgumentException e) {
    System.out.println("Неправильные аргументы, введенные в метод
multiply");
}
}
}

```

## Задание: Вариант 2

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

## Ход работы: Код программы файла Patient

```

import java.util.regex.Pattern;
public class Patient {
    private int id;

```

```

private String name;
private String surname;
private String lastname;
private String address;
private String phone;
private int cardNumber;
private String diagnosis;

public Patient() {
}

public Patient(int id, String name, String surname, String lastname,
String address, String phone, int cardNumber, String diagnosis) throws
Exception{

    if (id < 0) {
        throw new Exception("Исключение для пациента: неправильный id");
    }
    if (cardNumber <= 0) {
        throw new Exception("Исключение для пациента: неправильный номер
карты");
    }
    if ((name.equals("")) || (surname.equals("")) ||
(lastname.equals(""))) {
        throw new Exception("Исключение для пациента: пустое
имя/фамилия/фамилия!");
    }
    if (!Pattern.matches("^8-9\\d{2}-\\d{3}-\\d{2}-\\d{2}", phone)) {
        throw new Exception("Исключение для пациента: неправильный номер
телефона");
    }
    if (diagnosis.equals("")) {
        throw new Exception("Исключение для пациента: нет диагноза");
    }

    this.id = id;
    this.name = name;
    this.surname = surname;
    this.lastname = lastname;
    this.address = address;
    this.phone = phone;
    this.cardNumber = cardNumber;
    this.diagnosis = diagnosis;
}

public int getId() {
    return id;
}

public void setId(int id) {
    this.id = id;
}

public String getName() {
    return name;
}

public void setName(String name) {
    this.name = name;
}

public String getSurname() {
    return surname;
}

```



```

    }

    public void setSurname(String surname) {
        this.surname = surname;
    }

    public String getLastname() {
        return lastname;
    }

    public void setLastname(String lastname) {
        this.lastname = lastname;
    }

    public String getAddress() {
        return address;
    }

    public void setAddress(String address) {
        this.address = address;
    }

    public String getPhone() {
        return phone;
    }

    public void setPhone(String phone) {
        this.phone = phone;
    }

    public int getCardNumber() {
        return cardNumber;
    }

    public void setCardNumber(int cardNumber) {
        this.cardNumber = cardNumber;
    }

    public String getDiagnosis() {
        return diagnosis;
    }

    public void setDiagnosis(String diagnosis) {
        this.diagnosis = diagnosis;
    }

    @Override
    public String toString() {
        return "Пациент{" +
            "id=" + id +
            ", Имя='" + name + '\'' +
            ", Фамилия='" + surname + '\'' +
            ", Отчество='" + lastname + '\'' +
            ", Адрес='" + address + '\'' +
            ", Телефон='" + phone + '\'' +
            ", № карты=" + cardNumber +
            ", Диагнес='" + diagnosis + '\'' +
            '}';
    }
}

```

## Код программы файла MainPatient

```
import java.util.ArrayList;

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

        Patient[] patientsArray = new Patient[0];
        try {
            patientsArray = createPatientsArray();
        } catch (Exception e) {
            e.printStackTrace();
        }
        System.out.println("Пациенты:");
        for (Patient p: patientsArray) {
            System.out.println(p);
        }

        Patient[] patientsWithCOVID = chooseByDiagnosis(patientsArray,
"COVID-19");
        System.out.println();
        System.out.println("Пациент с COVID-19:");
        for (Patient p: patientsWithCOVID) {
            System.out.println(p);
        }
        Patient[] patientsInRange = chooseByCardNumber(patientsArray, 130,
140);
        System.out.println();
        System.out.println("Пациенты с № карты в диапазоне 130...140:");
        for (Patient p: patientsInRange) {
            System.out.println(p);
        }

    }

    private static Patient[] createPatientsArray() throws Exception{
        Patient p1 = new Patient(1,"Сергей", "Иванов", "Игоревич", "Дом №12",
"8-968-374-26-47", 132, "Астма");
        Patient p2 = new Patient(2,"Григорий", "Кайдмен", "Петрович", "Дом №
125", "8-969-375-27-74", 148, "Глоукома");
        Patient p3 = new Patient(3,"Владимир", "Костромин", "Олегович", "Дом
№ 13/2", "8-977-234-86-07", 119, "Диабет");
        Patient p4 = new Patient(4,"Алексей", "Миронов", "Александрович",
"Дом 5", "8-978-306-36-43", 135, "COVID-19");
        Patient p5 = new Patient(5,"Анна", "Миронова", "Генадевна", "Дом 5",
"8-961-333-28-17", 138, "COVID-19");
        return new Patient[]{p1, p2, p3, p4, p5};
    }

    private static Patient[] chooseByDiagnosis(Patient[] patientsArray,
String diagnosis){
        ArrayList<Patient> newPatientsArray = new ArrayList<>();
        for (int i = 0; i < patientsArray.length; i++) {
            if(patientsArray[i].getDiagnosis().equals(diagnosis)){
                newPatientsArray.add(patientsArray[i]);
            }
        }
        return (Patient[]) newPatientsArray.toArray(new
Patient[newPatientsArray.size()]);
    }

    private static Patient[] chooseByCardNumber(Patient[] patientsArray, int
startBound, int endBound){
```

```

        ArrayList<Patient> newPatientsArray = new ArrayList<>();
        for (int i = 0; i < patientsArray.length; i++) {
            if(patientsArray[i].getCardNumber() >= startBound &&
patientsArray[i].getCardNumber() <= endBound){
                newPatientsArray.add(patientsArray[i]);
            }
        }
        return (Patient[]) newPatientsArray.toArray(new
Patient[newPatientsArray.size()]);
    }
}

```

## Код программы файла Abiturient

```

import java.util.ArrayList;
import java.util.regex.*;

public class Abiturient {
    private int id;
    private String name;
    private String surname;
    private String lastname;
    private String address;
    private String phone;
    private ArrayList<Integer> marks;

    public Abiturient() {
    }

    public Abiturient(int id, String name, String surname, String lastname,
String address, String phone, ArrayList<Integer> marks) throws Exception {
        if (id < 0) {
            throw new Exception("Исключение для Абитуриента: неправильный
id");
        }
        if ((name.equals("")) || (surname.equals("")) ||
(lastname.equals("")))) {
            throw new Exception("Исключение для Абитуриента: пустое
имя/фамилия/фамилия");
        }
        if (!Pattern.matches("^8-9\\d{2}-\\d{3}-\\d{2}-\\d{2}", phone)) {
            throw new Exception("Исключение для Абитуриента: wrong phone
number!");
        }
        for(int x : marks){
            if((x < 2) || (x > 5)){
                throw new Exception("Abiturient exception : неправильный
номер телефона");
            }
        }
        this.id = id;
        this.name = name;
        this.surname = surname;
        this.lastname = lastname;
        this.address = address;
        this.phone = phone;
        this.marks = marks;
    }

    public int getId() {
        return id;
    }
}

```

```

public void setId(int id) {
    this.id = id;
}

public String getName() {
    return name;
}

public void setName(String name) {
    this.name = name;
}

public String getSurname() {
    return surname;
}

public void setSurname(String surname) {
    this.surname = surname;
}

public String getLastName() {
    return lastname;
}

public void setLastName(String lastname) {
    this.lastname = lastname;
}

public String getAddress() {
    return address;
}

public void setAddress(String address) {
    this.address = address;
}

public String getPhone() {
    return phone;
}

public void setPhone(String phone) {
    this.phone = phone;
}

public ArrayList<Integer> getMarks() {
    return marks;
}

public void setMarks(ArrayList<Integer> marks) {
    this.marks = marks;
}

@Override
public String toString() {
    return "Abiturient{" +
        "id=" + id +
        ", name='" + name + '\'' +
        ", surname='" + surname + '\'' +
        ", lastname='" + lastname + '\'' +
        ", address='" + address + '\'' +
        ", phone='" + phone + '\'' +
        ", marks=" + marks +
        '}';
}

```

## Код программы файла MainForAbiturient

```
import java.util.*;

public class MainForAbiturient {

    public static void main(String[] args) {

        Abiturient a1 = null;
        try {
            a1 = new Abiturient(1, "Иван", "Иванов", "Иванович", "Дом № 5",
"8968-374-26-47", new ArrayList<Integer>(Arrays.asList(3, 2, 5)));
        } catch (Exception e) {
            e.printStackTrace();
        }
        Abiturient a2 = null;
        try {
            a2 = new Abiturient(2, "Petr", "Petrov", "", "House 3", "8-969-
375-27-74", new ArrayList<Integer>(Arrays.asList(4, 4, 5)));
        } catch (Exception e) {
            e.printStackTrace();
        }
        Abiturient a3 = null;
        try {
            a3 = new Abiturient(3, "Dmitry", "Smirnov", "Ivanovich", "House
9", "8-977-234-86-07", new ArrayList<Integer>(Arrays.asList(5, 0, 5)));
        } catch (Exception e) {
            e.printStackTrace();
        }
        Abiturient a4 = null;
        try {
            a4 = new Abiturient(4, "Ivan", "Smirnov", "Andreevich", "House
5", "8-978-306-36-43", new ArrayList<Integer>(Arrays.asList(3, 2, 4)));
        } catch (Exception e) {
            e.printStackTrace();
        }
        Abiturient a5 = null;
        try {
            a5 = new Abiturient(5, "Alexander", "Ivanov", "Ilich", "House
11", "8-961-333-28-17", new ArrayList<Integer>(Arrays.asList(5, 5, 5)));
        } catch (Exception e) {
            e.printStackTrace();
        }

        Abiturient[] abiturientsArray = new Abiturient[]{a1, a2, a3, a4, a5};

        Abiturient[] abiturientsWithNeuds =
chooseWithNeuds(abiturientsArray);
        System.out.println();
        System.out.println("Abiturients with neuds:");
        for (Abiturient a : abiturientsWithNeuds) {
            System.out.println(a);
        }

        Abiturient[] abiturientsWithHigherAVG =
chooseHigherAVGMark(abiturientsArray, 4f);
        System.out.println();
        System.out.println("Abiturients with average mark higher then 4:");
        for (Abiturient a : abiturientsWithHigherAVG) {
            System.out.println(a);
        }

        Abiturient[] abiturientsBestN = chooseBest(abiturientsArray, 2);
```

```

        System.out.println();
        System.out.println("Best 2 abiturients:");
        for (Abiturient a : abiturientsBestN) {
            System.out.println(a);
        }

    }

    private static Abiturient[] chooseWithNeuds(Abiturient[]
abiturientsArray) {
        ArrayList<Abiturient> newAbiturientsArray = new ArrayList<>();
        for (int i = 0; i < abiturientsArray.length; i++) {
            try {
                if (abiturientsArray[i].getMarks().contains(2)) {
                    newAbiturientsArray.add(abiturientsArray[i]);
                }
            } catch (Exception e) {
            }
        }
        return (Abiturient[]) newAbiturientsArray.toArray(new
Abiturient[newAbiturientsArray.size()]);
    }

    private static Abiturient[] chooseHigherAVGMark(Abiturient[]
abiturientsArray, float mark) {
        ArrayList<Abiturient> newAbiturientsArray = new ArrayList<>();
        for (int i = 0; i < abiturientsArray.length; i++) {
            try {
                float avg = 0;
                for (Integer m : abiturientsArray[i].getMarks()) {
                    avg += m;
                }
                avg = avg / abiturientsArray[i].getMarks().size();
                if (avg > mark) {
                    newAbiturientsArray.add(abiturientsArray[i]);
                }
            } catch (Exception e) {
            }
        }
        return (Abiturient[]) newAbiturientsArray.toArray(new
Abiturient[newAbiturientsArray.size()]);
    }

    private static Abiturient[] chooseBest(Abiturient[] abiturientsArray,
Integer n) {
        ArrayList<Abiturient> newAbiturientsArray = new ArrayList<>();

        SortedMap<Float, ArrayList<Abiturient>> map = new TreeMap<>();
        for (int i = 0; i < abiturientsArray.length; i++) {
            try {
                float avg = 0;
                for (Integer m : abiturientsArray[i].getMarks()) {
                    avg += m;
                }
                avg = avg / abiturientsArray[i].getMarks().size();

                if (map.containsKey(avg)) {
                    map.get(avg).add(abiturientsArray[i]);
                } else {
                    map.put(avg, new ArrayList<>());
                    map.get(avg).add(abiturientsArray[i]);
                }
            } catch (Exception e) {
            }
        }
    }

```

```

    }
}
System.out.println(map);

int j = 0;
int avg_num = -1;
List<Float> floatList = new ArrayList<Float>(map.keySet());
Collections.reverse(floatList);
while (j < n) {
    avg_num++;
    for (int i = 0; i < map.get(floatList.get(avg_num)).size(); i++)
    {
newAbiturientsArray.add(map.get(floatList.get(avg_num)).get(i));
        j++;
    }
}

return (Abiturient[]) newAbiturientsArray.toArray(new
Abiturient[newAbiturientsArray.size()]);
}
}

```

### Задание: Вариант 3

В следующих заданиях требуется 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.

3. В каждой строке найти слова, начинающиеся с гласной буквы.

Ход работы: Файл The\_original.txt

```

Косы растрепаны, страшная, белая,
Бегаёт, бегаёт, резвая, смелая.
Темная ночь молчаливо пугается,
Шалями тучек луна закрывается.
Ветер-певун с завываньем кликуш
Мчится в лесную дремучую глушь.
Роща грозит еловыми пиками,
Прячутся совы с пугливыми криками.
Машет колдунья руками костлявыми.
Звезды моргают из туч над дубравами.
Серьгами змеи под космы привешены,
Кружится с вьюгой страшно и бешено.
Пляшет колдунья под звон сосняка.
С черною дрожью плывут облака.

```

## Код программы файла Роем

```
import java.io.File;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.nio.charset.StandardCharsets;
import java.nio.file.Files;
import java.nio.file.Path;
import java.nio.file.Paths;
import java.nio.file.StandardOpenOption;
import java.util.Collections;
import java.util.List;
import java.util.Locale;
import java.util.Scanner;

public class Poem {
    public static void main(String[] args) {
        File inp_file = new
File("C:\\Users\\maste\\IdeaProjects\\Lab5\\src\\The_original.txt");
        Path out_file_path =
Paths.get("C:\\Users\\maste\\IdeaProjects\\Lab5\\src\\Task1.txt");
        File out_file = new
File("C:\\Users\\maste\\IdeaProjects\\Lab5\\src\\Task1.txt");
        String vowels = "аоеиыуёя";

        if(out_file.delete()) {
            try {
                out_file.createNewFile();
            } catch (IOException e) {
                e.printStackTrace();
            }
        }

        Scanner scanner = null;
        try {
            scanner = new Scanner(inp_file);
        } catch (FileNotFoundException e) {
            e.printStackTrace();
        }
        while(scanner.hasNextLine()) {
            String line = scanner.nextLine();

            line = line.replaceAll("\\pP", "");
            String[] words = line.split(" ");
            String out = "";
            for(String word : words){
                String first_letter = word.length() > 1 ? word.substring(0,
1) : word;
                first_letter = first_letter.toLowerCase(Locale.ROOT);
                if (vowels.contains(first_letter)) {
                    out = out.concat(word).concat(" ");
                }
            }

            try {
                Files.write(out_file_path, Collections.singleton(out),
StandardCharsets.UTF_8, StandardOpenOption.APPEND);
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
    }
}
```



## Файл Task1.txt

еловыми

из

и

облака

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

### Ход работы: Код программы Роем2

```
import java.io.File;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.nio.charset.StandardCharsets;
import java.nio.file.Files;
import java.nio.file.Path;
import java.nio.file.Paths;
import java.nio.file.StandardOpenOption;
import java.util.Collections;
import java.util.List;
import java.util.Locale;
import java.util.Scanner;

public class Poem2 {

    public static void main(String[] args) {
        File inp_file = new
File("C:\\Users\\maste\\IdeaProjects\\Lab5\\src\\The_original.txt");
        Path out_file_path =
Paths.get("C:\\Users\\maste\\IdeaProjects\\Lab5\\src\\Task2.txt");
        File out_file = new
File("C:\\Users\\maste\\IdeaProjects\\Lab5\\src\\Task2.txt");

        if(out_file.delete()) {
            try {
                out_file.createNewFile();
            } catch (IOException e) {
                e.printStackTrace();
            }
        }

        Scanner scanner = null;
        try {
            scanner = new Scanner(inp_file);
        } catch (FileNotFoundException e) {
            e.printStackTrace();
        }
    }
}
```

```

        while(scanner.hasNextLine()) {
            String line = scanner.nextLine();

            line = line.replaceAll("\\pP", "");
            String[] arrayWords = line.split(" ");
            String out = "";

            for (int i = 0; i < arrayWords.length-1; i++) {
                if((arrayWords[i].charAt(arrayWords[i].length()-1)) ==
(arrayWords[i+1].charAt(0))) {
                    out = out.concat(arrayWords[i]).concat("
").concat(arrayWords[i + 1]);
                }
            }

            try {
                Files.write(out_file_path, Collections.singleton(out),
StandardCharsets.UTF_8, StandardOpenOption.APPEND);
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
    }
}

```

## Файл Task2.txt

над дубравами

## Задание: Вариант 4

При выполнении следующих заданий для вывода результатов создавать новую директорию и файл средствами класса File

3. Прочитать текст Java-программы и в каждом слове длиннее двух символов все строчные символы заменить прописными.

Ход работы: Код программы Task\_4\_3

```

import java.io.File;
import java.io.FileNotFoundException;
import java.io.IOException;

```

```

import java.nio.charset.StandardCharsets;
import java.nio.file.Files;
import java.nio.file.Path;
import java.nio.file.Paths;
import java.nio.file.StandardOpenOption;
import java.util.Collections;
import java.util.Locale;
import java.util.Scanner;
public class Task_4_3 {
    public static void main(String[] args) {

        File inp_file = new
File("C:\\Users\\maste\\IdeaProjects\\Lab5\\src\\Program1.java");

        File out_file = new
File("C:\\Users\\maste\\IdeaProjects\\Lab5\\src\\Java_program_result.txt");

        out_file.delete();

        try {
            out_file.createNewFile();
        } catch (IOException e) {
            e.printStackTrace();
        }

        Path out_file_path = Path.of(out_file.getPath());

        Scanner scanner = null;
        try {
            scanner = new Scanner(inp_file);
        } catch (FileNotFoundException e) {
            e.printStackTrace();
        }
        while(scanner.hasNextLine()) {
            String line = scanner.nextLine();

            //line = line.replaceAll("\\pP", " ");
            String[] words = line.split("[;!();\\ #.,:\\"]");

            for(String word : words){
                if (word.length() > 2) {
                    String word_upper = word.toUpperCase(Locale.ROOT);
                    line = line.replace(word, word_upper);
                }
            }

            try {
                Files.write(out_file_path, Collections.singleton(line),
StandardCharsets.UTF_8, StandardOpenOption.APPEND);
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
    }
}

```

```
IMPORT JAVA.UTIL.SCANNER;

PUBLIC CLASS PROGRAM1 {

    PUBLIC STATIC VOID MAIN(STRING[] ARGS) {
        INT n;
        SCANNER SCANNER = NEW SCANNER(SYSTEM.in);
        SYSTEM.OUT.PRINT("ВВЕДИТЕ КОЛ-ВО СТРОК" + ": ");
        n = SCANNER.NEXTINT();
        SCANNER.NEXTLINE();
        STRING[] STRING = NEW STRING[N];
        FOR (INT i = 0; i < n; I++) {
            SYSTEM.OUT.PRINT(" HOMEP СТРОКИ " + (I+1) + ": ");
            STRING[I] = SCANNER.NEXTLINE(); }

        SYSTEM.OUT.PRINTLN (" ВЫВОД БЕЗ ПЕРЕХОД на НОВУЮ
СТРОКУ " + ": ");
        FOR (INT i = 0; i < n; I++) {
            SYSTEM.OUT.PRINT (STRING[I]);
        }
        SYSTEM.OUT.PRINTLN();

        SYSTEM.OUT.PRINTLN (" ВЫВОД ПЕРЕХОД на НОВУЮ СТРОКУ
" + ": ");
        FOR (INT i = 0; i < n; I++) {
            SYSTEM.OUT.PRINTLN (STRING[I]);
        }
    }
}
```

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

Ход работы: Файл Student.txt

```
Иванов 8 8 8 8 8 8
Петров 6 7 8 4 6 8
Сидоров 7 7 7 7 8 7
Смирнов 6 6 6 6 8 8
Павлов 8 8 8 8 8 8
```

## Код программы Task\_4\_4

```
import java.io.File;
import java.io.IOException;
import java.io.PrintWriter;
import java.util.Scanner;
public class Task_4_4 {

    public static void main(String[] args) throws IOException {

        Scanner sc = new Scanner(new
File("C:\\Users\\maste\\IdeaProjects\\Lab5\\src\\Students.txt"));
        PrintWriter pw = new PrintWriter(new
File("C:\\Users\\maste\\IdeaProjects\\Lab5\\src\\Students_result.txt"));

        String[] student;

        while (sc.hasNextLine()) {
            student = sc.nextLine().split("\\s+");
            double sum = 0;
            for (int i = 1; i < student.length; i++) {
                sum += Integer.parseInt(student[i]);
            }
            if (sum / (student.length - 1) > 7) {
                pw.println(student[0].toUpperCase());
            }
        }

        sc.close();
        pw.close();

    }

}
```

## Файл Students\_result.txt

```
ИВАНОВ
СИДОРОВ
ПАВЛОВ
```

**Вывод:** лабораторная работа была выполнена в соответствии с заданием и полученные верные результаты работ программ