



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

Вариант 10

Название: Работа с исключениями и файлами

Дисциплина: Языки программирования для работы с большими данными

Студент

ИУ6-23М

(Группа)

(Подпись, дата)

И.С. Коломенский

(И.О. Фамилия)

Преподаватель

(Подпись, дата)

П.В. Степанов

(И.О. Фамилия)

Москва, 2022

Цель работы:

Получение навыков обработки исключений в Java и навыков работы с файлами в Java.

Выполнение:

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.

1. Определить класс Вектор размерности n . Реализовать методы сложения, вычитания, умножения, инкремента, декремента, индексирования. Определить массив из m объектов. Каждую из пар векторов передать в методы, возвращающие их скалярное произведение и длины. Вычислить и вывести углы между векторами.
2. Определить класс Булева матрица (BoolMatrix) размерности $(n \times m)$. Класс должен содержать несколько конструкторов. Реализовать методы для логического сложения (дизъюнкции), умножения и инверсии матриц. Реализовать методы для подсчета числа единиц в матрице и упорядочения строк в лексикографическом порядке.

Задание 2:

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

1. Student: id, Фамилия, Имя, Отчество, Дата рождения, Адрес, Телефон, Факультет, Курс, Группа. Создать массив объектов. Вывести: а) список студентов заданного факультета; б) списки студентов для каждого факультета и курса; в) список студентов, родившихся после заданного года;

d) список учебной группы.

2. Train: Пункт назначения, Номер поезда, Время отправления, Число мест (общих, купе, плацкарт, люкс). Создать массив объектов. Вывести: а) список поездов, следующих до заданного пункта назначения; б) список поездов, следующих до заданного пункта назначения и отправляющихся после заданного часа; с) список поездов, отправляющихся до заданного пункта назначения и имеющих общие места.

Задание 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.

1. В каждой строке найти и удалить заданную подстроку.
2. 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.

Задание 4:

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

1. Прочитать текст Java-программы и все слова `public` в объявлении атрибутов и методов класса заменить на слово `private`.
2. Ввести из текстового файла, связанного с входным потоком, последовательность строк. Выбрать и сохранить `m` последних слов в каждой из последних `n` строк

Листинг файла BoolMatrix.java

```
package com.company;
import java.util.Arrays;
import java.util.Random;
public class BoolMatrix {
    int n;
    int m;
    int true_count;
    boolean[][] matrix;

    public BoolMatrix(int n, int m) throws Exception{
        if(n<=0||m<=0)
            throw new Exception("Размер не может быть равно 0 или отрицательным");
        Random r = new Random();
        this.n = n;
        this.m = m;
        this.matrix = new boolean[n][m];
        if(matrix==null)
            throw new StackOverflowError("Памяти не хватает!");
        for (int i = 0; i != n; i++) {
            for (int j = 0; j != m; j++) {
                this.matrix[i][j] = r.nextBoolean();
                if (this.matrix[i][j] == true)
                {
                    this.true_count++;
                }
            }
        }
    }
    public void m_print()
    {
        System.out.printf("\nМатрица размерностью %d на %d\n", this.n, this.m);
        for (int i=0; i!=this.n; i++)
        {
            for (int j=0;j!=this.m;j++)
            {
                System.out.printf("%2b\t", this.matrix[i][j]);
            }
            System.out.println("");
        }
        System.out.printf("В матрице содержится %d элемента(ов) равных 1\n",
this.true_count);
    }
    public static void m_summ(BoolMatrix m1, BoolMatrix m2)
    {
        System.out.println("\nЛогическое сложение матриц");
        for (int i=0;i!=m1.n;i++)
        {
            for (int j=0; j!=m1.m; j++)
            {
                System.out.printf("%1b\t", m1.matrix[i][j] | m2.matrix[i][j]);
            }
            System.out.println("");
        }
    }
    public static void m_umn(BoolMatrix m1, BoolMatrix m2)
    {
        System.out.println("\nЛогическое умножение матриц");
        for (int i=0;i!=m1.n;i++)
        {
            for (int j=0; j!=m1.m; j++)
            {
                System.out.printf("%1b\t", m1.matrix[i][j] & m2.matrix[i][j]);
            }
            System.out.println("");
        }
    }
}
```

```

}
public void m_invers()
{
    System.out.println("\nИнверсия матрицы\nБыло:");
    this.m_print();
    for (int i=0; i!=this.n; i++)
    {
        for (int j=0; j!=this.m; j++)
        {
            this.matrix[i][j] = !this.matrix[i][j];
        }
    }

    System.out.println("Стало:");
    this.tr_count();
    this.m_print();
}

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

public void b_sort()
{
    System.out.println("\nСортировка матрицы в лексикографическом порядке\nБыло:");
    this.m_print();
    boolean[] cash = new boolean[this.m];
    int count_1=0, count_2=0;
    for (int ii=0;ii!=this.n;ii++) {
        for (int i = 0; i != this.n - 1; i++) {
            for (int j = 0; j != this.m; j++) {
                if (this.matrix[i][j]) {
                    count_1 += (int) Math.pow(this.m - j, 2);
                }
                if (this.matrix[i + 1][j]) {
                    count_2 += (int) Math.pow(this.m - j, 2);
                }
            }
            if (count_1 < count_2) {
                for (int n = 0; n != this.m; n++) {
                    cash[n] = this.matrix[i][n];
                }
                for (int n = 0; n != this.m; n++) {
                    this.matrix[i][n] = this.matrix[i + 1][n];
                }
                for (int n = 0; n != this.m; n++) {
                    this.matrix[i + 1][n] = cash[n];
                }
            }
            count_1 = 0;
            count_2 = 0;
        }
    }
    System.out.println("Стало:");
    this.m_print();
}
}

```

Листинг файла Library.java

```
package com.company;
import java.util.Random;

public class Library {
    private static final int PHONE_NUMBER_LENGTH = 11;
    private static final int PHONE_CARD_LENGTH = 16;
    public static String[] first_name = {"Denis", "Kirill", "Nikolay", "Danil",
"Nasty", "Elena", "Lukas", "Garry", "Luk"};
    public static String[] last_name = {"Buksha", "Golikov", "Alimov"};
    public static String[] patronymic_name = {"AndreeWitch", "AlexeeWitch"};
    public static String[] faculty = {"Faculty of Computer Science", "Faculty of
History", "Faculty of Economics", "Faculty of Linguistics"};

    public static void print_array(int[][] mas_int){
        for (int i=0;i < mas_int.length;i++,System.out.println(" ")){
            for (int j=0;j < mas_int[i].length;j++){
                if (mas_int [i][j] >= 0)
                    System.out.print(" "+mas_int [i][j]+" ");
                else
                    System.out.print(mas_int [i][j]+" ");
            }
        }
    }

    public static void print_array(String[][] mas_int){
        for (int i=0;i < mas_int.length;i++,System.out.println(" ")){
            for (int j=0;j < mas_int[i].length;j++){
                System.out.print(mas_int[i][j]);
            }
        }
    }

    public static String getDate(){
        int month, year, day;
        Random call = new Random();
        month = call.nextInt(11)+1;
        year = 1990 + (int) (Math.random() * 20);
        day = call.nextInt(28)+1;

        String calendar = day+"."+month+"."+year;
        return calendar;
    }

    public static String getRandomPhone() {
        String s = "123456789";
        StringBuffer phoneNumber = new StringBuffer();

        for (int i = 0; i < PHONE_NUMBER_LENGTH; i++) {
            phoneNumber.append(s.charAt(new Random().nextInt(s.length())));
        }
        return phoneNumber.toString();
    }

    public static String getRandomCard() {
        String s = "123456789";
        StringBuffer phoneNumber = new StringBuffer();
        for (int i = 0; i < PHONE_CARD_LENGTH; i++) {
            phoneNumber.append(s.charAt(new Random().nextInt(s.length())));
        }
        return phoneNumber.toString();
    }
}
```

Листинг файла MyVector.java

```
package com.company;

import java.util.Arrays;
public class MyVector {
    private int[][] vector_date_mas;
    private Object[] vector_mas;
    private int last_obj;

    public MyVector(){
        int n = 4;
        this.vector_date_mas = new int[2][n];

        if(this.vector_date_mas==null)//Если памяти не выделялось
            throw new StackOverflowError("Памяти не хватает");

        for (int i = 0; i < n; i++) {
            vector_date_mas[0][i] = (int) (Math.random() * 10);
            vector_date_mas[1][i] = (int) (Math.random() * 10);
        }
        this.vector_mas = new Object[n];
        if(this.vector_mas==null)//Если память не выделялась
            throw new StackOverflowError("Памяти не хватает");
        for (int i = 0; i < n; i++) {
            vector_mas[i] = new int[2];
            if(this.vector_mas[i]==null)//Если память не выделялась
                throw new StackOverflowError("Памяти не хватает");
            vector_date_mas[0][i] = (int) (Math.random() * 10);
            vector_date_mas[1][i] = (int) (Math.random() * 10);
        }
    }

    public MyVector(int n){
        this.vector_date_mas = new int[2][n];
        if(this.vector_date_mas==null)//Если память не выделялась
            throw new StackOverflowError("Памяти не хватает");
        for (int i = 0; i < n; i++) {
            vector_date_mas[0][i] = (int) (Math.random() * 10);
            vector_date_mas[1][i] = (int) (Math.random() * 10);
        }
        this.vector_mas = new Object[n];
        for (int i = 0; i < n; i++) {
            vector_mas[i] = new int[2];
            vector_date_mas[0][i] = (int) (Math.random() * 10);
            vector_date_mas[1][i] = (int) (Math.random() * 10);
        }
    }

    public int[][] getVector_date_mas() {
        return vector_date_mas;
    }

    @Override
    public String toString() {
        return "MyVector{" +
            "vector_date_mas=" + Arrays.toString(vector_date_mas) +
            '}';
    }

    public int[] get_sum(int x, int y){
        int[] sum = new int[2];
        if(sum==null)//Если память не выделялась
            throw new StackOverflowError("Памяти не хватает");
        sum[0] = vector_date_mas[0][x] + vector_date_mas[0][y];
        sum[1] = vector_date_mas[1][x] + vector_date_mas[1][y];
        return sum;
    }
}
```

```

}

public int[] get_subtract(int x, int y){
    int[] sum = new int[2];
    if(sum==null)//Если память не выделялась
        throw new StackOverflowError("Памяти не хватает");
    sum[0] = vector_date_mas[0][x] - vector_date_mas[0][y];
    sum[1] = vector_date_mas[1][x] - vector_date_mas[1][y];
    return sum;
}

public int[] get_inc(int x){
    int[] sum = new int[2];
    if(sum==null)//Если память не выделялась
        throw new StackOverflowError("Памяти не хватает");

    sum[0] = vector_date_mas[0][x] +1;
    sum[1] = vector_date_mas[1][x] +1;
    return sum;
}

public int[] get_dec(int x){
    int[] sum = new int[2];
    if(sum==null)//Если память не выделялась
        throw new StackOverflowError("Памяти не хватает");

    sum[0] = vector_date_mas[0][x] -1;
    sum[1] = vector_date_mas[1][x] -1;
    return sum;
}

public int get_index(int x, int y){
    int index = vector_date_mas[x][y];
    return index;
}

public int get_scalar(int x, int y){
    int[] sum = new int[2];
    if(sum==null)//Если память не выделялась
        throw new StackOverflowError("Памяти не хватает");
    int scalar = 0;
    sum[0] = vector_date_mas[0][x] * vector_date_mas[0][y];
    sum[1] = vector_date_mas[1][x] * vector_date_mas[1][y];
    scalar = sum[0] + sum[1];
    return scalar;
}

public double get_length(int x){
    double leng = 0;
    leng =
Math.sqrt(vector_date_mas[0][x]*vector_date_mas[0][x]+vector_date_mas[1][x]*vector_date
_mas[1][x]);
    return leng;
}

public double get_corner(int x, int y){
    double scal = get_scalar(x , y);
    double modul1 = get_length(x);
    double modul2 =get_length(y);
    double corne = scal/(modul1*modul2);
    return corne;
}

public boolean get_collin(int x, int y){
    boolean collin;
    if (vector_date_mas[0][y] != 0 && vector_date_mas[1][y] != 0)
        if ((vector_date_mas[0][x]/vector_date_mas[0][y]) ==

```



```

(vector_date_mas[1][x]/vector_date_mas[1][y])) {
    collin = true;
} else
    collin = false;
else
    collin = false;
return collin;
}
public boolean get_ortogonal(int x, int y){
    boolean ortogonal;
    if (get_scalar(x,y) == 0)
        ortogonal = true;
    else
        ortogonal = false;
    return ortogonal;
}
}

```

Листинг файла Student.java

```

package com.company;

import java.util.Date;

public class Student {
    private int id;
    private String first_name;
    private String last_name;
    private String patronymic_name;
    private String date;
    private String address;
    private String phone;
    private String faculty;
    private int course;
    private int group;
    private long card;
    private long bank_;

    public Student(int x){
        this.id = x;
        this.first_name = Library.first_name[(int) (Math.random()*9)];
        this.last_name = Library.last_name[(int) (Math.random()*3)];
        this.patronymic_name = Library.patronymic_name[(int) (Math.random()*2)];
        this.date = Library.getDate();
        this.address = "Address" + (int) (Math.random()*9);
        this.phone = Library.getRandomPhone();
        this.faculty = Library.faculty[(int) (Math.random()*4)];
        this.course = 1 + (int) (Math.random()*6);
        this.group = (int) (Math.random()*4);
        this.card = Long.parseLong(Library.getRandomCard());
        this.bank_ = Long.parseLong(Library.getRandomCard());
    }

    public void setDate(String date) {
        this.date = date;
    }

    @Override
    public String toString() {
        return "Student{" +
            "id=" + id +
            ", first_name='" + first_name + '\'' +
            ", last_name='" + last_name + '\'' +
            ", patronymic_name='" + patronymic_name + '\'' +
            ", date='" + date + '\'' +
            ", address='" + address + '\'' +

```

```

        ", phone='" + phone + '\'' +
        ", faculty='" + faculty + '\'' +
        ", course=" + course +
        ", group=" + group +
        ", card=" + card +
        ", bank_" + bank_ +
        '}'';
    }

    public void setCard(long card) {
        this.card = card;
    }

    public void setBank_(long bank_) {
        this.bank_ = bank_;
    }

    public long getCard() {
        return card;
    }

    public void setCard(int card) {
        this.card = card;
    }

    public long getBank_() {
        return bank_;
    }

    public void setBank_(int bank_) {
        this.bank_ = bank_;
    }

    public int getId() {
        return id;
    }

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

    public String getFirst_name() {
        return first_name;
    }

    public void setFirst_name(String first_name) {
        this.first_name = first_name;
    }

    public String getLast_name() {
        return last_name;
    }

    public void setLast_name(String last_name) {
        this.last_name = last_name;
    }

    public String getPatronymic_name() {
        return patronymic_name;
    }

    public void setPatronymic_name(String patronymic_name) {
        this.patronymic_name = patronymic_name;
    }

    public String getDate() {

```

```

        return date;
    }

    public void setDate(Date date) {
        this.date = String.valueOf(date);
    }

    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 String getFaculty() {
        return faculty;
    }

    public void setFaculty(String faculty) {
        this.faculty = faculty;
    }

    public int getCourse() {
        return course;
    }

    public void setCourse(int course) {
        this.course = course;
    }

    public int getGroup() {
        return group;
    }

    public void setGroup(int group) {
        this.group = group;
    }
}

```

Листинг файла Train.java

```

package com.company;
import java.util.Random;
//Генерируем исключение то где это нужно
public class Train {
    //Пункт назначения, Номер поезда, Время отправления, Число мест (общих, купе,
    плацкарт, люкс)
    String arrival;
    int train_no;
    int time_out;
    int total_place, kupe_place, pla_place, luxe_place;
    public Train() {
        String[] arriaval_places = {"Омск", "Рязань", "Москва", "Питер", "Надым",
"Тула", "Кольма"};
        Random r = new Random();
        if(r==null)
            throw new StackOverflowError("Память не хватает");
        this.arrival = arriaval_places[r.nextInt(6)];
    }
}

```

```

        this.train_no = r.nextInt(10000) + 1;
        this.time_out = r.nextInt(23) + 1;
        this.kupe_place = r.nextInt(40);
        this.pla_place = r.nextInt(100) + 1;
        this.luxe_place = r.nextInt(25);
        this.total_place = this.kupe_place + this.pla_place + this.luxe_place;
    }

    public void Tr_print() {
        System.out.println("Инфрмация о поезде:");
        System.out.printf("Пункт названчения - %s\nНомер поезда - %d\nЧас отправления - %d\nОбщее число мест - %d\nМеста купе - %d\n" +
            "Места плацкорт - %d\nМеста люкс - %d\n\n", this.arrival,
this.train_no, this.time_out, this.total_place, this.kupe_place, this.pla_place,
this.luxe_place);
    }

    public static Train[] create_mass(int n) {
        Train[] mass = new Train[n];
        if(mass==null)
            throw new StackOverflowError("Память не хватает");
        for (int i = 0; i != n; i++) {
            mass[i] = new Train();
        }
        return mass;
    }

    public static void select(Train[] mass, int n, String arrival, int time_out, int
total_place)
    {
        for (int i=0; i!=n; i++)
        {
            if (time_out == 0 && total_place == 0)
            {
                if (mass[i].arrival.equals(arrival))
                {
                    mass[i].Tr_print();
                }
            }else if (time_out > 0 && total_place == 0)
            {
                if (mass[i].time_out > time_out && mass[i].arrival.equals(arrival))
                {
                    mass[i].Tr_print();
                }
            } else
            {
                if (mass[i].total_place > total_place &&
mass[i].arrival.equals(arrival))
                {
                    mass[i].Tr_print();
                }
            }
        }
    }
}

```

Листинг файла Main1.java

```

package com.company;
import java.util.Arrays;
public class Main1 {

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

```

```

{
    System.out.println("\nВариант 1 №1");
    int n = 6;
    MyVector vector = new MyVector(n);
    //Вывод Сложения
    System.out.println("\nСложение");
    System.out.println(Arrays.toString(vector.get_sum(0, 1)));
    //Вывод вычитания
    System.out.println("\nВычитание");
    System.out.println(Arrays.toString(vector.get_subtract(2, 3)));
    //Вывод инкремента вектора
    System.out.println("\nИнкремент");
    System.out.println(Arrays.toString(vector.get_inc(0)));
    //Вывод декремента вектора
    System.out.println("\nДекремент");
    System.out.println(Arrays.toString(vector.get_dec(0)));
    //Вывод индекса указанного вектора
    System.out.println("\nИндекс");
    System.out.println(vector.get_index(1, 3));
    //Вывод скалярного произведения
    System.out.println("\nСкалярное произведение");
    System.out.println(vector.get_scalar(0, 1));
    //Вывод длины вектора
    System.out.println("\nДлина вектора");
    System.out.println(vector.get_length(1));
    //Угол векторов
    System.out.println("\nУгол вектора");
    System.out.println(vector.get_corner(0, 1));

    System.out.println("\nВариант 1 №10");
    BoolMatrix m_1 = new BoolMatrix(6,6);
    m_1.m_print();
    BoolMatrix m_2 = new BoolMatrix(6,6);
    m_2.m_print();
    BoolMatrix.m_summ(m_1,m_2);
    BoolMatrix.m_umn(m_1,m_2);
    m_1.m_invers();
    m_1.b_sort();
}
catch (Exception e)
{
    System.out.println(e.toString());
}
catch (StackOverflowError mem)
{
    System.out.println(mem.toString());
}
}
}

```

Листинг файла Main2.java

```

package com.company;
import java.util.Arrays;

public class Main2 {

    public static void main(String[] args) {
        try {
            System.out.println("\nВариант 2 №1");
            int n = 6;
            int counter = 0;
            Student[] students = new Student[n];
            if(students==null)
                throw new StackOverflowError("Не возможно выделить память");
            for (int i = 0; i < n; i++) {

```

```

        students[i] = new Student(i);
    }
    System.out.println(Arrays.toString(students));
    //список студентов заданного факультета;
    System.out.println("Список студентов заданного факультета: ");
    for (int i = 0; i < n; i++) {
        if (students[i].getFaculty().equals("Faculty of Computer Science")) {
            System.out.println(students[i]);
            counter = 1;
        }
    }
    if (counter == 0)
        System.out.println("По заданному факультету студентов НЕТ");
    //списки студентов для каждого факультета и курса
    System.out.println("|_|_|_|_|_|_|_|_|");
    System.out.println("|_|_|_|_|_|_|_|_|");
    System.out.println("Списки студентов для каждого факультета и курса");

    for (int i = 0; i < Library.faculty.length; i++) {
        for (int j = 1; j < 7; j++) {
            System.out.println("Факультет: " + Library.faculty[i] + " , Курс: " +
j);

            counter = 0;
            for (int k = 0; k < n; k++) {
                if (students[k].getFaculty().equals(Library.faculty[i]) &&
students[k].getCourse() == j) {
                    System.out.println(students[k]);
                    counter = 1;
                }
            }
            if (counter == 0)
                System.out.println("По заданному факультету и курсу студентов
НЕТ");
        }
        counter = 0;
    }
    System.out.println("|_|_|_|_|_|_|_|_|");
    System.out.println("|_|_|_|_|_|_|_|_|");
    //список студентов, родившихся после заданного года
    counter = 0;
    System.out.println("Список студентов, родившихся после заданного года: Будет
после 1999 ");
    for (int i = 0; i < n; i++) {
        if
(Integer.parseInt(students[i].getDate().substring(students[i].getDate().length()-4)) >
1999) {
            System.out.println(students[i]);
            counter = 1;
        }
    }
    if (counter == 0)
        System.out.println("студентов старше НЕТ");
    System.out.println("|_|_|_|_|_|_|_|_|");
    System.out.println("|_|_|_|_|_|_|_|_|");
    //список учебной группы.
    System.out.println("список учебной группы: Например Информатический факультет
1 курса 1 группы");
    students[1].setFaculty(Library.faculty[0]);
    students[1].setCourse(1);
    students[1].setGroup(1);
    for (int i = 0; i < n; i++) {
        if (students[i].getFaculty().equals("Faculty of Computer Science") &&
students[i].getCourse() == 1 && students[i].getGroup() == 1) {
            System.out.println(students[i]);
            counter = 1;
        }
    }

```

```

    }

    System.out.println("\nВариант 2 №10");
    n = 8;
    Train[] masstr;
    masstr = Train.create_mass(n);
    for (int i=0; i!=n;i++)
    {
        masstr[i].Tr_print();
    }
    System.out.println("Запрос (а) список поездов, следующих до заданного пункта назначения");
    Train.select(masstr,n,"Москва",0,0);
    System.out.println("Запрос (б) список поездов, следующих до заданного пункта назначения и отправляющихся после заданного часа");
    Train.select(masstr,n,"Питер",10,0);
    System.out.println("Запрос (с) список поездов, отправляющихся до заданного пункта назначения и имеющих общие места");
    Train.select(masstr,n,"Надым",0,100);
}
catch (StackOverflowError mem)
{
    System.out.println(mem.toString());
}
catch (Exception ex)
{
    System.out.println(ex.toString());
}
}
}

```

Листинг файла Main3_1.java

```

package com.company;

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

public class Main3_1 { // edit configuration -> build and run -> set program arguments to
    // C://Users//Wingo//IdeaProjects//BigDataLanguages// input.txt output.txt
    public static String DelSubString(String source,String del)
    {
        return source.replace(del,"");
    }

    public static void main(String[] args) {
        try
        {
            System.out.println(args[1]);
            if(args.length<3)
                throw new IOException("Недостаточно параметров");
            File inp=new File(args[0]+args[1]);
            File out=new File(args[0]+args[2]);
            if(!inp.isFile())
                throw new IOException("Ошибка: входной файл не существует");
            if(!out.isFile()) // file does not exist
            {
                out.createNewFile();
            }
            FileReader reader=new FileReader(inp);
            Scanner sc=new Scanner(reader);
            FileWriter writer=new FileWriter(out);
            Scanner cin=new Scanner(System.in);

```

```

        System.out.println("Введите подстроку: ");
        String sub= cin.next();
        while(sc.hasNextLine())
        {
            String str=sc.nextLine();
            str=DelSubString(str,sub);
            writer.write(str+"\n");
        }
        reader.close();
        writer.close();
    }
    catch (IOException ex)
    {
        System.out.println(ex.toString());
    }
}
}

```

Листинг файла Main3_2.java

```

package com.company;

import java.io.*;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.stream.Collectors;
import java.util.stream.IntStream;

public class Main3_2 {
    public static void main(String[] args) throws IOException {
        File file = new File("test.txt");
        FileOutputStream fout = new FileOutputStream(file);

        String greetings = "1 2 3 4 5 6\r\n7 8 9 1 2 3\n4 5 6 7 8 9\n";

        fout.write(greetings.getBytes());

        fout.close();

        FileInputStream fileInputStream = new FileInputStream(file);
        String zap = "";

        ArrayList<ArrayList<Integer>> mass = new ArrayList<>();
        ArrayList<ArrayList<String>> mass_str = new ArrayList<>();
        BufferedReader br = new BufferedReader(new FileReader(file));
        String line = "";
        while ((line = br.readLine()) != null) {
            ArrayList<String> mass_str_c = new ArrayList<>();
            mass.add(IntStream.of(Arrays.stream(line.split(" "))
                .mapToInt(Integer::valueOf).toArray()).boxed().collect(Collectors.toCollection(ArrayList::new)));
            mass_str_c.addAll(Arrays.asList(line.split(" ")));
            mass_str.add(mass_str_c);
        }
        System.out.println("Двумерный массив исходной матрицы: ");
        System.out.println(mass);
        System.out.println("Размерность x - "+mass.size()+" y - "+mass.get(0).size());
        System.out.println("");

        String new_str_file = "";
        for (int i=0; i!=mass_str.size(); i++){
            new_str_file+= String.join(" ", mass_str.get(i));
            new_str_file+="\n";
        }
        System.out.println("Исходная матрица: ");
        System.out.println(new_str_file);
    }
}

```



```

        System.out.println("Результат поворота матрицы на 90 градусов по часовой
стрелке: ");
        ArrayList<ArrayList<Integer>> mass_r = new ArrayList<>();
        for (int i = 0; i < mass.get(0).size(); i++) {
            ArrayList<Integer> mass_line = new ArrayList<>();
            for (int j = mass.size() - 1; j >= 0; j--) {
                mass_line.add(mass.get(j).get(i));
                System.out.print(mass.get(j).get(i) + "\t");
            }
            mass_r.add(mass_line);
            System.out.println();
        }

        System.out.print(mass_r);
        System.out.println();
    }
}

```

Листинг файла Main4_1.java

```

package com.company;

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

public class Main4_1 {
    public static void main(String[] args) {
        try
        {
            System.out.println("Введите текст программы (конец текста пустая строка)");
            String program="";
            String line="";
            Scanner sc=new Scanner(System.in);
            while(sc.hasNextLine())
            {
                line= sc.nextLine();
                if(line.compareTo("")==0)//Если пустая строка
                    break;
                program=program+line+"\n";
            }
            program=program.replaceAll("public","private");
            System.out.println(program);
            File file=new File("MyDerictoriya");
            file.mkdirs();
            if(!file.exists())
                throw new IOException("Не возможно создать дерикторию");
            FileWriter writer=new FileWriter(file.toString()+"//output.txt");
            writer.write(program);
            writer.close();
        }
        catch (IOException ex)
        {
            System.out.println(ex.toString());
        }
    }
}

```

Листинг файла Main4_2.java

```

package com.company;

import java.io.*;
import java.util.ArrayList;
import java.util.Arrays;

```

```

import java.util.Scanner;
import java.util.stream.Collectors;
import java.util.stream.IntStream;

public class Main4_2 {
    public static void main(String[] args) throws IOException {
        File file = new File("test.txt");

        ArrayList<ArrayList<Integer>> mass = new ArrayList<>();
        BufferedReader br = new BufferedReader(new FileReader(file));
        String line = "";
        while ((line = br.readLine()) != null) {
            mass.add(IntStream.of(Arrays.stream(line.split(" ")
            ")).mapToInt(Integer::valueOf).toArray()).boxed().collect(Collectors.toCollection(Array
            List::new)));
        }

        Scanner in = new Scanner(System.in);
        System.out.println(mass);
        System.out.print("Введите m (слов) - ");
        int m = in.nextInt();
        System.out.print("Введите n (строк) - ");
        int n = in.nextInt();
        ArrayList<ArrayList<Integer>> mass_str_l_nm = new ArrayList<>();
        for (int i=mass.size()-n; i!=mass.size(); i++){
            ArrayList<Integer> mass_s = new ArrayList<>();
            for (int j=mass.get(i).size()-m; j!=mass.get(i).size(); j++){
                mass_s.add(mass.get(i).get(j));
            }
            mass_str_l_nm.add(mass_s);
        }
        System.out.println(mass_str_l_nm);
    }
}

```

```

Вариант 1 №1

Сложение
[17, 12]

Вычитание
[-2, 7]

Инкремент
[10, 6]

Декремент
[8, 4]

Индекс
1

Скалярное произведение
107

Длина вектора
10.63014581273465

Угол вектора
0.9776685014774172

```

Рисунок 1 - Результат выполнения кода решения задачи 1, подзадачи 1

```

Вариант 1 №10
Инверсия матрицы
Было:
Матрица размерностью 6 на 6
false true true false true false
true false true true true true
true true true false false true
true false true true true true
false true false true false true
false false false true true true
В матрице содержится 23 элемента(ов) равных 1
Матрица размерностью 6 на 6
false false true false true false
true true true false false false
true false true false false true
true true true true false false
true true false false true true
false true true false false true
В матрице содержится 19 элемента(ов) равных 1
Логическое сложение матриц
false true true false true false
true true true true true true
true true true false false true
true true true true true true
true true false true true true
false true true true true true
Логическое умножение матриц
false false true false true false
true false true false false false
true false true false false true
true false true true false false
false true false false false true
false false false false false true
Стало:
Матрица размерностью 6 на 6
true true true false false false
true false true false true false
true false false true true true
false true false false false true
false true false false false true
false false false true true true
В матрице содержится 13 элемента(ов) равных 1
Стало:
Матрица размерностью 6 на 6
true true true false false false
true false true false true false
true false false true false true
false true false false false true
false true false false false true
false false false true true true
В матрице содержится 13 элемента(ов) равных 1
Process finished with exit code 0
Сортировка матрицы в лексикографическом порядке
Было:
Матрица размерностью 6 на 6
true false false true false true
false true false false false false
false false false true true false
false true false false false false
true false true false true false
true true true false false false
В матрице содержится 13 элемента(ов) равных 1
Стало:

```

Рисунок 2 - Результат выполнения кода решения задачи 1, подзадачи 2


```

Вариант 2 №10
Информация о поезде:
Пункт назначения - Омск
Номер поезда - 5980
Час отправления - 5
Общее число мест - 145
Места купе - 31
Места плацкарт - 93
Места люкс - 21

Информация о поезде:
Пункт назначения - Москва
Номер поезда - 2526
Час отправления - 4
Общее число мест - 102
Места купе - 20
Места плацкарт - 73
Места люкс - 9

Информация о поезде:
Пункт назначения - Омск
Номер поезда - 2462
Час отправления - 14
Общее число мест - 50
Места купе - 16
Места плацкарт - 26
Места люкс - 8

Информация о поезде:
Пункт назначения - Рязань
Номер поезда - 5534
Час отправления - 13
Общее число мест - 72
Места купе - 15
Места плацкарт - 51
Места люкс - 6

Информация о поезде:
Пункт назначения - Питер
Номер поезда - 7062
Час отправления - 11
Общее число мест - 74
Места купе - 39
Места плацкарт - 27
Места люкс - 8

Информация о поезде:
Пункт назначения - Омск
Номер поезда - 8238
Час отправления - 20
Общее число мест - 45
Места купе - 28
Места плацкарт - 4
Места люкс - 13

Информация о поезде:
Пункт назначения - Москва
Номер поезда - 9965
Час отправления - 3
Общее число мест - 106
Места купе - 18
Места плацкарт - 79
Места люкс - 9

Информация о поезде:
Пункт назначения - Питер
Номер поезда - 7062
Час отправления - 11
Общее число мест - 74
Места купе - 39
Места плацкарт - 27
Места люкс - 8

Запрос (б) список поездов, следующих до заданного пункта назначения и отправляющихся после заданного часа
Информация о поезде:
Пункт назначения - Питер
Номер поезда - 7062
Час отправления - 11
Общее число мест - 74
Места купе - 39
Места плацкарт - 27
Места люкс - 8

Запрос (с) список поездов, отправляющихся до заданного пункта назначения и имеющих общие места
Process finished with exit code 0

```

Рисунок 4 - Результат выполнения кода решения задачи 2, подзадачи 2

```

Введите подстроку:
Wello

Process finished with exit code 0

1 Hello
2 Wello sub my
3 sub my three

1 Hello
2 sub my
3 sub my three

```

Рисунок 5 - Результат выполнения кода решения задачи 3, подзадачи 1

```

Двумерный массив исходной матрицы:
[[1, 2, 3, 4, 5, 6], [7, 8, 9, 1, 2, 3], [4, 5, 6, 7, 8, 9]]
Размерность x - 3 y - 6

Исходная матрица:
1 2 3 4 5 6
7 8 9 1 2 3
4 5 6 7 8 9

Результат поворота матрицы на 90 градусов по часовой стрелке:
4 7 1
5 8 2
6 9 3
7 1 4
8 2 5
9 3 6
[[4, 7, 1], [5, 8, 2], [6, 9, 3], [7, 1, 4], [8, 2, 5], [9, 3, 6]]

```

Рисунок 6 - Результат выполнения кода решения задачи 3, подзадачи 2

```

Введите текст программы (конец текста пустая строка)
class Person{
public int a;
public int b;
public String s;
}

class Person{
private int a;
private int b;
private String s;
}

```

Рисунок 7 - Результат выполнения кода решения задачи 4, подзадачи 1

```

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

```

Рисунок 8 - Результат выполнения кода решения задачи 4, подзадачи 2

Ссылка на программное решение:

Программное решение представлено в репозитории распределённой системы управления версиями Git:

<https://github.com/Wingo11/BigDataLanguages/tree/Lab5/src>

Вывод:

При выполнении лабораторной работы были получены навыки обработки исключений в Java и навыки работы с файлами в Java.