



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 № 3

Вариант 10

Название: Классы, наследование и полиморфизм

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 работы с классами Java, исследование механизмов наследования и полиморфизма.

Выполнение:

Задание 1:

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:

Создать классы, спецификации которых приведены ниже. Определить конструкторы и методы `setТип()`, `getТип()`, `toString()`. Определить дополнительно методы в классе, создающем массив объектов. Задать критерий выбора данных и вывести эти данные на консоль.

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; г) 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; в) список поездов, отправляющихся до заданного пункта

назначения и имеющих общие места

Задание 3:

Создать приложение, удовлетворяющее требованиям, приведенным в задании. Аргументировать принадлежность классу каждого создаваемого метода и корректно переопределить для каждого класса методы equals(), hashCode(), toString().

1. Создать объект класса Текстовый файл, используя класс Файл. Методы: создать, переименовать, вывести на консоль содержимое, дополнить, удалить.
2. Создать объект класса Год, используя классы Месяц, День. Методы: задать дату, вывести на консоль день недели по заданной дате, рассчитать количество дней, месяцев в заданном временном промежутке.

Задание 4:

1. Система Факультатив. Преподаватель объявляет запись на Курс. Студент записывается на Курс, обучается и по окончании Преподаватель выставляет Оценку, которая сохраняется в Архиве. Студентов, Преподавателей и Курсов при обучении может быть несколько.
2. Система Железнодорожная касса. Пассажир делает Заявку на станцию назначения, время и дату поездки. Система регистрирует Заявку и осуществляет поиск подходящего Поезда. Пассажир делает выбор Поезда и получает Счет на оплату. Администратор вводит номера Поездов, промежуточные и конечные станции, цены.

Листинг файла Cours.java

```
package com.company;
import java.io.File;
import java.io.FileWriter;
import java.io.IOException;
import java.nio.file.Files;
import java.nio.file.Paths;
import java.util.Arrays;

public class Cours {
    int idCours;
    Teach[] teaches = new Teach[2];
    public Cours(int idCours) {
        this.idCours = idCours;
        for (int i = 0; i < 2; i++) {
            teaches[i] = new Teach(i, idCours);
        }
    }
}
```

```

    }
}

public void createfile () {
    String pathname = "/Users/Wingo/Downloads/arxivForCours"+idCours+".txt";
    File file = new File(pathname);
    String data = this.toString();
    try {
        Files.write(Paths.get(pathname), data.getBytes());
    } catch (IOException e) {
        e.printStackTrace();
    }
}

@Override
public String toString() {
    return "Cours{" +
        "idCours=" + idCours +
        ", teaches=" + Arrays.toString(teaches) +
        '}';
}
}

```

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.ArrayList;
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];
        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 MyVector(int n) {
        this.vector_date_mas = new int[2][n];
        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];
    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];
    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];
    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];
    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];
    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 com.company.Library;

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 Studentt.java

```

package com.company;

public class Studentt {
    int idStudent, idTeach, idCours, score;
    String FIO;
    public Studentt(int idCours, int idTeach, int idStudent) {
        this.idStudent = idStudent;
        this.idTeach = idCours;
        this.idTeach = idTeach;
        this.FIO = idTeach + " " + idStudent + " " + idStudent;
        this.score = (int) (Math.random() * 6);
    }
}

```

```

@Override
public String toString() {
    return "Student{" +
        "idStudent=" + idStudent +
        ", idTeach=" + idTeach +
        ", idCours=" + idCours +
        ", score=" + score +
        ", FIO='" + FIO + '\'' +
        '}';
}
}

```

Листинг файла Teach.java

```

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

public class Teach {
    int idTeach;
    int idCours;
    String FIO;
    Studentt[] students = new Studentt[2];
    public Teach(int idTeach, int idCours) {
        this.idTeach = idTeach;
        this.idCours = idCours;
        this.FIO = idTeach+" "+idTeach+" "+idTeach;
        for (int i = 0; i < 2; i++) {
            students[i] = new Studentt(idCours,idTeach,i);
        }
    }
    @Override
    public String toString() {
        return "Teach{" +
            "idTeach=" + idTeach +
            ", idCours=" + idCours +
            ", FIO='" + FIO + '\'' +
            ", students=" + Arrays.toString(students) +
            '}';
    }
}

```

Листинг файла TextFile.java

```

package com.company;
import java.io.File;
import java.io.*;
import java.nio.file.Files;
import java.nio.file.Paths;
import java.util.Objects;

public class TextFile extends File {
    String pathname;

    //Методы: создать, переименовать, вывести на консоль содержимое, дополнить,
    удалить.
    void create(){
        String data = "Это тестовая строка для записи в файл";
        try {
            Files.write(Paths.get(pathname), data.getBytes());
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
    void rename(){
        File renameFile = new File("/Users/Wingo/Downloads/renamefiles.txt");
        if(this.renameTo(renameFile)){
            System.out.println("Файл переименован успешно");
        }
    }
}

```

```

    }else{
        System.out.println("Файл не был переименован");
    }
    //this.pathname = "/Users/Wingo/Downloads/renamefiles.txt";
}
void print () throws IOException {
    FileReader reader = new FileReader(this);
    int c;
    while ((c= reader.read())!=-1){
        System.out.print((char)c);
    }
    reader.close();
}
void write() throws IOException{
    FileWriter writer = new FileWriter(this,true);
    String dopText = ". Дополнение файла прошло успешно";
    writer.write(dopText);
    writer.flush();
    writer.close();
    System.out.println("Файл дополнен.");
}
void delFile(){
    if(this.delete()){
        System.out.println("Файл удален.");
    }else System.out.println("Файла не удален.");
}

public TextFile(String pathname) {
    super(pathname);
    this.pathname = pathname;
    create();
}

@Override
public String toString() {
    return "TextFile{" +
        "pathname='" + pathname +
        '\'';
}

@Override
public boolean equals(Object o) {
    if (this == o) return true;
    if (!(o instanceof TextFile)) return false;
    if (!super.equals(o)) return false;
    TextFile textFile = (TextFile) o;
    return Objects.equals(pathname, textFile.pathname);
}

@Override
public int hashCode() {
    return Objects.hash(super.hashCode(), pathname);
}
}

```

Листинг файла Var1.java

```

package com.company;

import java.util.Arrays;
import java.util.Random;

public class Var1
{
    public static class BoolMatrix {
        int n;
    }
}

```

```

int m;
int true_count;
boolean[][] matrix;

public BoolMatrix(int n, int m) {
    Random r = new Random();
    this.n = n;
    this.m = m;
    this.matrix = new boolean[n][m];
    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(Var1.BoolMatrix m1, Var1.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(Var1.BoolMatrix m1, Var1.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();
}

}

public static void main(String[] args) {
    System.out.println("\nВариант 1 №1");
    int n = 6;
    MyVector vector = new MyVector(n);
    //vector.sum(x, y);
    Library.print_array(vector.getVector_date_mas());
    //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");
Var1.BoolMatrix m_1 = new Var1.BoolMatrix(6,6);
m_1.m_print();
Var1.BoolMatrix m_2 = new Var1.BoolMatrix(6,6);
m_2.m_print();
Var1.BoolMatrix.m_summ(m_1,m_2);
Var1.BoolMatrix.m_umn(m_1,m_2);
m_1.m_invers();
m_1.b_sort();
}
}

```

Листинг файла Var2.java

```

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

public class Var2 {
    public static 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", "Колыма"};
            Random r = new Random();
            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 плацкарт - %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];
        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();
                }
            }
        }
    }

}

public static void main(String[] args) {
    System.out.println("\nВариант 2 №1");
    int n = 6;
    int counter = 0;
    Student[] students = new Student[n];
    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);
}
}

```

Листинг файла Var3.java

```

package com.company;

import java.io.IOException;
import java.util.*;
import java.time.format.DateTimeFormatter;
import java.time.format.DateTimeFormatterBuilder;
import java.text.SimpleDateFormat;
import java.text.DateFormat;
import java.time.LocalDate;
import java.time.Period;

import java.util.concurrent.TimeUnit;

public class Var3 {
    public static void main(String[] args) {
        System.out.println("\nВариант 3 №10"); // DONE
        year year2022 = new year(2022);
        for (int i=0; i!=1;i++){
            String name;
            int number, count;
            Scanner in = new Scanner(System.in);
            System.out.print("Введите название месяца - ");
            name = in.nextLine();
            System.out.print("Введите номер месяца - ");
            number = in.nextInt();
            System.out.print("Введите кол-во дней в месяце - ");
            count = in.nextInt();
            month new_month = new month(name, number);
            for (int j=1; j<=count;j++){
                day new_day = new day(j);
                new_month.add_day(new_day);
            }
            year2022.add_new_month(new_month);
        }
        System.out.println(year2022.toString());
        System.out.println("");
        System.out.println("Вывести день неделя заданной даты");
        Scanner in = new Scanner(System.in);
        int m_n, d_n;
        Scanner in1 = new Scanner(System.in);
        System.out.print("Введите номер месяца - ");
        m_n = in1.nextInt();
        System.out.print("Число месяца - ");
        d_n = in1.nextInt();
        Calendar show_data = new GregorianCalendar(2022, m_n-1, d_n);
        System.out.println(show_data.getTime());
        Calendar show_data1 = new GregorianCalendar(2022, m_n-1, d_n);
        String begin, end;
        System.out.print("Введите начальную дату формата dd.MM.yyyy - ");
        begin = in.nextLine();
        System.out.print("Введите конечную дату формата dd.MM.yyyy - ");
        end = in.nextLine();
        DateTimeFormatter formatter = DateTimeFormatter.ofPattern("dd.MM.yyyy");
        LocalDate startDate = LocalDate.parse(begin, formatter);
        LocalDate endDate = LocalDate.parse(end, formatter);
        Period period = Period.between(startDate, endDate);
    }
}

```

```

System.out.println("Прошло лет - "+period.getYears());
System.out.println("Прошло месяцев - "+period.getMonths());
System.out.println("Прошло дней - "+period.getDays());

System.out.println("-----");
System.out.println("Вариант 3 №1"); // DONE
TextFile file = new TextFile("/Users/Wingo/Downloads/files.txt");
System.out.println(file);
file.rename();
System.out.println(file);
try {
    file.write();
    file.print();
} catch (IOException e){
    System.out.println("no print/ no write");
}
System.out.println(" ");
file.delFile();
}

public static class day{
    int daytime_str;
    String notes;

    public day(int daytime_str) {
        this.daytime_str = daytime_str;
    }

    public day(int daytime_str, String notes) {
        this.daytime_str = daytime_str;
        this.notes = notes;
    }

    public day(){}

    @Override
    public boolean equals(Object o) {
        if (this == o) return true;
        if (o == null || getClass() != o.getClass()) return false;
        day day = (day) o;
        return daytime_str == day.daytime_str && Objects.equals(notes, day.notes);
    }

    @Override
    public int hashCode() {
        return Objects.hash(daytime_str, notes);
    }

    @Override
    public String toString() {
        return "day{" +
            "daytime_str=" + daytime_str +
            ", notes='" + notes + '\'' +
            '}';
    }
}

public static class month extends day{
    String month_name;
    int month_number;
    String notes;
    ArrayList<day> days;

    public month(String month_name, int month_number, String notes, ArrayList<day>
days) {
        this.month_name = month_name;
        this.month_number = month_number;
    }
}

```

```

        this.notes = notes;
        this.days = days;
    }

    public month(String month_name, int month_number) {
        this.month_name = month_name;
        this.month_number = month_number;
        this.days = new ArrayList<>();
    }

    public void add_day(day newday) {
        this.days.add(newday);
    }

    public month(){}

    @Override
    public boolean equals(Object o) {
        if (this == o) return true;
        if (o == null || getClass() != o.getClass()) return false;
        if (!super.equals(o)) return false;
        month month = (month) o;
        return Objects.equals(month_name, month.month_name) &&
Objects.equals(notes, month.notes) && Objects.equals(days, month.days);
    }

    @Override
    public int hashCode() {
        return Objects.hash(super.hashCode(), month_name, notes, days);
    }

    @Override
    public String toString() {
        return "\nmonth{" +
            "month_name='" + month_name + '\'' +
            ", notes='" + notes + '\'' +
            ", days=" + days.toString() +
            '}';
    }
}

public static class year extends month{
    int year_number;
    String notes;
    ArrayList<month> monthes;

    public year(int year_number) {
        this.year_number = year_number;
        this.monthes = new ArrayList<>();
    }

    public void add_new_month(month new_month){
        this.monthes.add(new_month);
    }

    @Override
    public boolean equals(Object o) {
        if (this == o) return true;
        if (o == null || getClass() != o.getClass()) return false;
        if (!super.equals(o)) return false;
        year year = (year) o;
        return year_number == year.year_number && Objects.equals(notes,
year.notes);
    }

    @Override

```

```

        public int hashCode() {
            return Objects.hash(super.hashCode(), year_number, notes);
        }

        @Override
        public String toString() {
            return "year{" +
                "year_number=" + year_number +
                ", notes='" + notes + '\'' +
                ", monthes=" + monthes.toString() +
                '}';
        }
    }
}

```

Листинг файла Var4.java

```

package com.company;

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

public class Var4 {
    public static class passanger implements Serializable{
        private String FIO;
        private int balance;
        public ArrayList<passanger> pas_list;

        public passanger(String FIO, int balance) {
            this.FIO = FIO;
            this.balance = balance;
        }

        public passanger() {
            pas_list = new ArrayList<>();
        }

        @Override
        public String toString() {
            return "passanger{" +
                "FIO='" + FIO + '\'' +
                ", balance=" + balance +
                '}';
        }

        public String toString(int n) {
            if (n!=-1) {
                return "passanger{" +
                    "FIO='" + pas_list.get(n).FIO + '\'' +
                    ", balance=" + pas_list.get(n).balance +
                    '}';
            }else{
                return "Пассажир не найден";
            }
        }

        public void add_pas(String FIO, int balance){
            passanger cash = new passanger(FIO, balance);
            this.pas_list.add(cash);
        }

        public int search_pas(String name){
            for (int i=0;i!=this.pas_list.size();i++){
                if (this.pas_list.get(i).FIO.equals(name)){
                    return i;
                }
            }
        }
    }
}

```

```

    }
    return -1;
}

public void add_money_pass(int value){
    this.balance+=value;
}

}

public static class train implements Serializable{
    private String depin;
    private String depout;
    private int day;
    private int t_out;
    private int t_in;
    private int cost;
    ArrayList<train> train_list;

    public train(String depin, String depout, int day, int t_out, int t_in, int
cost) {
        this.depin = depin;
        this.depout = depout;
        this.day = day;
        this.t_out = t_out;
        this.t_in = t_in;
        this.cost = cost;
    }

    public train() {
        train_list = new ArrayList<>();
    }

    @Override
    public String toString() {
        return "train{" +
            "depin='" + depin + '\'' +
            ", depout='" + depout + '\'' +
            ", day=" + day +
            ", t_out=" + t_out +
            ", t_in=" + t_in +
            ", cost=" + cost +
            '}';
    }

    public String toString(int n) {
        if (n!=-1) {
            return "train{" +
                "depin='" + train_list.get(n).depin + '\'' +
                ", depout='" + train_list.get(n).depout + '\'' +
                ", day=" + train_list.get(n).day +
                ", t_out=" + train_list.get(n).t_out +
                ", t_in=" + train_list.get(n).t_in +
                ", cost=" + train_list.get(n).cost +
                '}';
        }else
        {
            return "Такого поезда нет";
        }
    }

    public void add_train(String depin, String depout, int day, int t_out, int
t_in, int cost){
        train cash = new train(depin, depout, day, t_out, t_in, cost);
        this.train_list.add(cash);
    }

    public int searchg_train(String depout, int t_out){

```

```

        for(int i=0; i!=this.train_list.size();i++){
            if (this.train_list.get(i).depout.equals(depout) &&
this.train_list.get(i).t_out==t_out){
                return i;
            }
        }
        return -1;
    }

}

public static void main(String[] args) throws IOException, ClassNotFoundException {
    System.out.println("Вариант 4 №10");
    passenger c_pass = new passenger();
    c_pass.add_pas("Вася", 10000);
    FileOutputStream pass_file_w = new FileOutputStream("passobj.txt");
    ObjectOutputStream passobj_w = new ObjectOutputStream(pass_file_w);
    passobj_w.writeObject(c_pass);
    passobj_w.close();

    train c_train = new train();
    c_train.add_train("Питер", "Москва", 4, 6, 20, 1200);
    FileOutputStream train_file_w = new FileOutputStream("trainobj.txt");
    ObjectOutputStream trainobj_w = new ObjectOutputStream(train_file_w);
    trainobj_w.writeObject(c_train);
    trainobj_w.close();

    FileInputStream pass_file = new FileInputStream("passobj.txt");
    ObjectInputStream passobj = new ObjectInputStream(pass_file);
    Object cash_i = passobj.readObject();
    passenger passangers = (passenger) cash_i;
    passobj.close();
    System.out.println("Пассажиры загружены");

    FileInputStream train_file = new FileInputStream("trainobj.txt");
    ObjectInputStream trainobj = new ObjectInputStream(train_file);
    Object cash_c = trainobj.readObject();
    train trains = (train) cash_c;
    trainobj.close();
    System.out.println("Поезда загружены");
    while (true) {
        Scanner in_1 = new Scanner(System.in);
        System.out.println("Вы пассажир или администратор? 1 - пассажир, 2 -
администратор, 3 - выйти");
        int who = in_1.nextInt();
        String p_name="";
        int p_id = 0;

        while (who != 3 && who != 55 && who != 66) {
            Scanner in = new Scanner(System.in);
            switch (who) {
                case 1:
                    System.out.println("Пассажир, введите своё имя");
                    p_name = in.nextLine();
                    p_id = passangers.search_pas(p_name);
                    if (p_id != -1) {
                        System.out.println("Добро пожаловать!");
                        who = 55;
                    } else {
                        who = -1;
                    }
                    System.out.println(passangers.toString(p_id));
                    break;
                case 2:
                    System.out.println("Введите пароль");

```

```

        if (in.nextLine().equals("123")) {
            System.out.println("Успешно");
            who = 66;
        } else {
            System.out.println("Пароль введён не верно");
            who = -1;
        }
        break;
    case 3:
        who = 3;
    default:
        System.out.println("Такого пункта меню нет в списке");
        who = -1;
    }
    if (who == -1) {
        System.out.println("Вы пассажир или администратор? 1 - пассажир, 2 - администратор, 3 - выйти");
        who = in.nextInt();
    }
}

if (who != 3) {
    System.out.println("Добро пожаловать в систему покупки билетов");
    boolean check = true;
    while (check) {
        if (who == 55) {
            System.out.println("Пассажир, выбери пункт меню для совершения действий\n1 - купить билет, 2 - пополнить баланс, 3 - выйти");
            Scanner in_p = new Scanner(System.in);
            switch (Integer.parseInt(in_p.nextLine())) {
                case 1:
                    System.out.println("Введите город отправления и время отправления");

                    in_p.reset();
                    String city = in_p.nextLine();
                    int t_out = Integer.parseInt(in_p.nextLine());
                    System.out.println("Информация о поезде");
                    int tr_id = trains.searchg_train(city, t_out);
                    System.out.println(trains.toString(tr_id));
                    System.out.println("Купить билет на этот поезд? 1 - да, 2 - нет");

                    switch (in_p.nextLine()) {
                        case "1":
                            if (passangers.pas_list.get(p_id).balance >= trains.train_list.get(tr_id).cost) {
                                passangers.pas_list.get(p_id).balance -= trains.train_list.get(tr_id).cost;
                                System.out.println("Покупка совершена");
                            } else {
                                System.out.println("У вас недостаточно средств");
                            }
                            break;
                        case "2":
                            System.out.println("Отмена покупки");
                            break;
                        default:
                            System.out.println("Такого пункта меню нет");
                            break;
                    }
                    break;
                case 2:
                    System.out.println("Введите сумму пополнения");
                    int value = Integer.parseInt(in_p.nextLine());
                    passangers.pas_list.get(p_id).add_money_pass(value);

```

```

System.out.println(passangers.pas_list.get(p_id).toString());
        case 3:
            check = false;
            break;
        default:
            System.out.println("Такого пункта меню нет");
            break;
    }
    } else {
        System.out.println("Администратор, выбери пункт меню для
совершения действий\n1 - добавить поезд, 2 - посмотреть информацию о поезде, 3 -
выйти");

        Scanner in_a = new Scanner(System.in);
        switch (in_a.nextLine()) {
            case "1":
                System.out.println("Введите:");
                System.out.print("Место прибытия - ");
                String d_in = in_a.nextLine();
                System.out.print("Место отбытия - ");
                String d_out = in_a.nextLine();
                System.out.print("День отправки - ");
                int day = Integer.parseInt(in_a.nextLine());
                System.out.print("Время отправки - ");
                int t_out = Integer.parseInt(in_a.nextLine());
                System.out.print("Время прибытия - ");
                int t_in = Integer.parseInt(in_a.nextLine());
                System.out.print("Стоимость - ");
                int cost = Integer.parseInt(in_a.nextLine());
                trains.add_train(d_in, d_out, day, t_out, t_in, cost);
                System.out.println("Новый рейс успешно добавлен");

System.out.println(trains.train_list.get(trains.searchg_train(d_out,
t_out)).toString());

                break;
            case "2":
                System.out.println("Введите:");
                System.out.print("Место отбытия - ");
                d_out = in_a.nextLine();
                System.out.print("Время отправки - ");
                t_out = Integer.parseInt(in_a.nextLine());

System.out.println(trains.train_list.get(trains.searchg_train(d_out,
t_out)).toString());

                break;
            case "3":
                check = false;
                break;
            default:
                System.out.println("Такого пункта меню нет");
                break;
        }
    }
}

//passanger c_pass = new passanger();
//c_pass.add_pas("Вася", 10000);
//
//    FileOutputStream pass_file_w = new FileOutputStream("passobj.txt");
//    ObjectOutputStream passobj_w = new ObjectOutputStream(pass_file_w);
//    passobj_w.writeObject(passangers);
//    passobj_w.close();
//
//
//    //train c_train = new train();
//    //c_train.add_train("Питер", "Москва", 4, 6, 20, 1200);

```



```

//      FileOutputStream train_file_w = new FileOutputStream("trainobj.txt");
//      ObjectOutputStream trainobj_w = new ObjectOutputStream(train_file_w);
//      trainobj_w.writeObject(trains);
//      trainobj_w.close();
pass_file_w = new FileOutputStream("passobj.txt");
passobj_w = new ObjectOutputStream(pass_file_w);
passobj_w.writeObject(passangers);
passobj_w.close();

//train c_train = new train();
//c_train.add_train("Питер", "Москва", 4, 6, 20, 1200);
train_file_w = new FileOutputStream("trainobj.txt");
trainobj_w = new ObjectOutputStream(train_file_w);
trainobj_w.writeObject(trains);
trainobj_w.close();
}
}
}

```

Вариант 1 №1	Вариант 1 №10	Матрица размерностью 6 на 6
4 8 7 1 2 9	Матрица размерностью 6 на 6	true true true false false false
7 4 6 4 9 5	true true true false false false	false false true false true true
Сложение	false false true false true true	false true true true true true
[12, 11]	false true true true true true	true false true false true false
Вычитание	true false true false true false	false false false true false true
[6, 2]	false false false true false true	false false false false true true
Инкремент	false false false false true true	В матрице содержится 18 элемента(ов) равных 1
[5, 8]	true false true false true true	Стало:
Декремент	false false true false true false	Матрица размерностью 6 на 6
[3, 6]	В матрице содержится 20 элемента(ов) равных 1	false false false true true true
Индекс	Логическое сложение матриц	true true false true false false
4	true true true true true true	true true false true false false
Скалярное произведение	true true true false true true	true false false false false false
60	false false true false true true	false true false true false true
Длина вектора	false true true false false false	true true true false true false
8.94427190999916	false false false false true false	true true false true false false
Угол вектора	false false true false true false	true false false false false false
0.8320502943378437	Инверсия матрицы	false true false true false true
	Было:	false false false true true true
		В матрице содержится 18 элемента(ов) равных 1

Рисунки 1, 2, 3 - Результат выполнения кода решения задачи 1

```

prompt 2 WS
[Student(id=0, first_name='Lukas', last_name='Buksha', patronymic_name='Alexeevitch', date='25.1.1993', address='Address4', phone='88857278937', faculty='Faculty of Linguistics', course=1, group=3, card=8658633469557747, bank_='st
Список студентов заданного факультета:
По заданному факультету студентов НЕТ
[.] [.] [.] [.] [.] [.] [.] [.] [.] [.]
[.] [.] [.] [.] [.] [.] [.] [.] [.] [.]
Список студентов для каждого факультета и курса
Факультет: Faculty of Computer Science , Курс: 1
По заданному факультету и курсу студентов НЕТ
Факультет: Faculty of Computer Science , Курс: 2
По заданному факультету и курсу студентов НЕТ
Факультет: Faculty of Computer Science , Курс: 3
По заданному факультету и курсу студентов НЕТ
Факультет: Faculty of Computer Science , Курс: 4
По заданному факультету и курсу студентов НЕТ
Факультет: Faculty of Computer Science , Курс: 5
По заданному факультету и курсу студентов НЕТ
Факультет: Faculty of Computer Science , Курс: 6
По заданному факультету и курсу студентов НЕТ
Факультет: Faculty of History , Курс: 1
По заданному факультету и курсу студентов НЕТ
Факультет: Faculty of History , Курс: 2
По заданному факультету и курсу студентов НЕТ
Факультет: Faculty of History , Курс: 3
По заданному факультету и курсу студентов НЕТ
Факультет: Faculty of History , Курс: 4
По заданному факультету и курсу студентов НЕТ
Факультет: Faculty of History , Курс: 5
По заданному факультету и курсу студентов НЕТ
Факультет: Faculty of History , Курс: 6
По заданному факультету и курсу студентов НЕТ
Факультет: Faculty of Economics , Курс: 1
По заданному факультету и курсу студентов НЕТ
Факультет: Faculty of Economics , Курс: 2
По заданному факультету и курсу студентов НЕТ
Факультет: Faculty of Economics , Курс: 3
По заданному факультету и курсу студентов НЕТ
Факультет: Faculty of Economics , Курс: 4
Student(id=3, first_name='Lukas', last_name='Alimov', patronymic_name='Alexeevitch', date='16.7.2001', address='Address5', phone='93851581617', faculty='Faculty of Economics', course=4, group=2, card=9484898739848699, bank_='85977
Факультет: Faculty of Economics , Курс: 5
По заданному факультету и курсу студентов НЕТ
Факультет: Faculty of Economics , Курс: 6
По заданному факультету и курсу студентов НЕТ
Факультет: Faculty of Linguistics , Курс: 1
Student(id=0, first_name='Lukas', last_name='Buksha', patronymic_name='Alexeevitch', date='25.1.1993', address='Address4', phone='88857278937', faculty='Faculty of Linguistics', course=1, group=3, card=8658633469557747, bank_='st
Факультет: Faculty of Linguistics , Курс: 2
Student(id=4, first_name='Lukas', last_name='Alimov', patronymic_name='Andreevitch', date='23.4.2004', address='Address1', phone='11849185416', faculty='Faculty of Linguistics', course=2, group=1, card=5381864929863795, bank_='19
Факультет: Faculty of Linguistics , Курс: 3
По заданному факультету и курсу студентов НЕТ
Факультет: Faculty of Linguistics , Курс: 4
По заданному факультету и курсу студентов НЕТ
Факультет: Faculty of Linguistics , Курс: 5
Student(id=0, first_name='Elena', last_name='Golikov', patronymic_name='Alexeevitch', date='3.7.1995', address='Address8', phone='91237829815', faculty='Faculty of Linguistics', course=5, group=3, card=5861553348444251, bank_='14
Факультет: Faculty of Linguistics , Курс: 6
Student(id=1, first_name='Luk', last_name='Golikov', patronymic_name='Andreevitch', date='9.6.2003', address='Address5', phone='17128725522', faculty='Faculty of Linguistics', course=6, group=2, card=551273325263362, bank_='4768
[.] [.] [.] [.] [.] [.] [.] [.] [.] [.]
[.] [.] [.] [.] [.] [.] [.] [.] [.] [.]
Список студентов, родившихся после заданного года: Бывает после 1999
Student(id=3, first_name='Luk', last_name='Golikov', patronymic_name='Andreevitch', date='9.6.2003', address='Address5', phone='17128725522', faculty='Faculty of Linguistics', course=6, group=2, card=551273325263362, bank_='4768
Student(id=3, first_name='Lukas', last_name='Alimov', patronymic_name='Alexeevitch', date='16.7.2001', address='Address5', phone='93851581617', faculty='Faculty of Economics', course=4, group=2, card=9484898739848699, bank_='85977
Student(id=4, first_name='Lukas', last_name='Alimov', patronymic_name='Andreevitch', date='23.4.2004', address='Address1', phone='11849185416', faculty='Faculty of Linguistics', course=2, group=1, card=5381864929863795, bank_='19
[.] [.] [.] [.] [.] [.] [.] [.] [.] [.]
[.] [.] [.] [.] [.] [.] [.] [.] [.] [.]
список учебной группы: Например Информативский факультет 1 курса 1 группы
Student(id=1, first_name='Luk', last_name='Golikov', patronymic_name='Andreevitch', date='9.6.2003', address='Address5', phone='17128725522', faculty='Faculty of Computer Science', course=1, group=1, card=551273325263362, bank_

```

Рисунок 4 - Результат выполнения кода решения задачи 2, подзадачи 1

Вариант 2 №10	Информация о поезде:	Запрос (а) список поездов, следующих до заданного пункта назначения
Информация о поезде:	Пункт назначения - Москва	Информация о поезде:
Пункт назначения - Питер	Номер поезда - 9909	Пункт назначения - Москва
Номер поезда - 666	Час отправления - 3	Номер поезда - 9909
Час отправления - 9	Общее число мест - 89	Час отправления - 3
Общее число мест - 112	Места купе - 28	Общее число мест - 89
Места купе - 17	Места плацкарт - 43	Места купе - 28
Места плацкарт - 76	Места люкс - 18	Места плацкарт - 43
Места люкс - 19		Места люкс - 18
	Информация о поезде:	
Информация о поезде:	Пункт назначения - Рязань	Информация о поезде:
Пункт назначения - Тула	Номер поезда - 353	Пункт назначения - Москва
Номер поезда - 7410	Час отправления - 3	Номер поезда - 635
Час отправления - 21	Общее число мест - 59	Час отправления - 12
Общее число мест - 68	Места купе - 24	Общее число мест - 18
Места купе - 9	Места плацкарт - 32	Места купе - 0
Места плацкарт - 58	Места люкс - 3	Места плацкарт - 1
Места люкс - 1		Места люкс - 17
	Информация о поезде:	
Информация о поезде:	Пункт назначения - Москва	Запрос (б) список поездов, следующих до заданного пункта назначения и отправляющихся после заданного часа
Пункт назначения - Надым	Номер поезда - 635	Запрос (с) список поездов, отправляющихся до заданного пункта назначения и имеющих общие места
Номер поезда - 2783	Час отправления - 12	Информация о поезде:
Час отправления - 6	Общее число мест - 18	Пункт назначения - Надым
Общее число мест - 72	Места купе - 0	Номер поезда - 3496
Места купе - 37	Места плацкарт - 1	Час отправления - 18
Места плацкарт - 31	Места люкс - 17	Общее число мест - 117
Места люкс - 4		Места купе - 37
	Информация о поезде:	Места плацкарт - 68
Информация о поезде:	Пункт назначения - Надым	Места люкс - 12
Пункт назначения - Рязань	Номер поезда - 3496	
Номер поезда - 2586	Час отправления - 18	
Час отправления - 4	Общее число мест - 117	
Общее число мест - 132	Места купе - 37	
Места купе - 19	Места плацкарт - 68	
Места плацкарт - 91	Места люкс - 12	
Места люкс - 22		

Рисунок 5 - Результат выполнения кода решения задачи 2, подзадачи 2

```

Вариант 3 №10
Введите название месяца - май
Введите номер месяца - 5
Введите кол-во дней в месяце - 30
year{year_number=2022, notes='null', monthes=[
month{month_name='май', notes='null', days=[day{daytime_str=1, notes='null'}, day{daytime_str=2,

Вывести день неделя заданной даты
Введите номер месяца - 4
Число месяца - 4
Mon Apr 04 00:00:00 MSK 2022
Введите начальную дату формата dd.MM.yyyy - 01.01.2000
Введите конечную дату формата dd.MM.yyyy - 10.10.2001
Прошло лет - 1
Прошло месяцев - 9
Прошло дней - 9

```

Рисунок 6 - Результат выполнения кода решения задачи 3, подзадачи 1

```

Вариант 3 №1
TextFile{pathname='/Users/Wingo/Downloads/files.txt'}
Файл не был переименован
TextFile{pathname='/Users/Wingo/Downloads/files.txt'}
Файл дополнен.
Это тестовая строка для записи в файл. Дополнение файла прошло успешно
Файл удален.

```

Рисунок 7 - Результат выполнения кода решения задачи 3, подзадачи 2

```

Вариант 4 №10
Пассажиры загружены
Поезда загружены
Вы пассажир или администратор? 1 - пассажир, 2 - администратор, 3 - выйти
1
Пассажир, введите своё имя
Вася
Добро пожаловать!
passanger{FIO='Вася', balance=10000}
Добро пожаловать в систему покупки билетов
Пассажир, выбери пункт меню для совершения действий
1 - купить билет, 2 - пополнить баланс, 3 - выйти
1
Введите город отправления и время отправления
Москва
0
Информация о поезде
Такого поезда нет

```

Рисунок 8 - Результат выполнения кода решения задачи 4, подзадач 1 и 2

Ссылка на программное решение:

Программное решение представлено в репозитории распределённой системы управления версиями Git:

<https://github.com/Wingo11/BigDataLanguages/tree/Lab3/src>

Вывод:

При выполнении лабораторной работы были получены навыки работы с классами Java, были исследованы механизмы наследования и полиморфизма языка программирования Java.