|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **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 09.04.01/07 **Интеллектуальные системы анализа, обработки и интерпретации больших данных**

**Отчет**

|  |
| --- |
| **по лабораторной работе №3** |

**Название:**

Классы, наследование, полиморфизм

**Дисциплина:** Языка программирования для работы с большими данными

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент | ИУ6-23М |  |  | Д.Д. Капитонов |
|  | (Группа) |  | (Подпись, дата) | (И.О. Фамилия) |
|  |  |  |  |  |
| Преподаватель |  |  |  | П.В. Степанов |
|  |  |  | (Подпись, дата) | (И.О. Фамилия) |

Москва, 2022

**Вариант 1, номера 9 и 10.**

9. Определить класс Квадратное уравнение. Класс должен содержать несколько конструкторов. Реализовать методы для поиска корней, экстремумов, а также интервалов убывания/возрастания. Создать массив объектов и определить наибольшие и наименьшие по значению корни.

10. Определить класс Булева матрица (BoolMatrix) размерности (n x 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

**Код программы:**

*package dan.lab;*

*import com.sun.org.apache.xpath.internal.operations.Bool;*

*import java.util.Arrays;*

*import java.util.Random;*

*public class lab3\_1 {*

*public static class kv\_yr*

*{*

*int a,b,c;*

*double x1,x2;*

*public kv\_yr(int a, int b, int c)*

*{*

*this.a = a;*

*this.b = b;*

*this.c = c;*

*}*

*public kv\_yr(int x)*

*{*

*Random r = new Random();*

*this.a = r.nextInt(12);*

*this.b = r.nextInt(12);*

*this.c = r.nextInt(12);*

*this.x1 = x;*

*}*

*public void kv\_print()*

*{*

*System.out.printf("Уравнение имеет вид: %d\*x^2+%d\*x+%d=0\n",this.a, this.b,this.c);*

*}*

*public String kv\_korni()*

*{*

*double d,x1,x2;*

*String result;*

*d = Math.pow(this.b,2) - 4\*this.a\*this.c;*

*if (d<0)*

*{*

*result = "У уравнения отсутствуют корни";*

*this.x1 = 666;*

*this.x2 = 666;*

*} else if (d == 0)*

*{*

*x1 = (-b+Math.sqrt(d))/(2\*this.a);*

*result = "Корень уравнения = " + x1;*

*this.x1 = x1;*

*this.x2 = this.x1;*

*} else*

*{*

*this.x1 = (-b+Math.sqrt(d))/(2\*this.a);*

*this.x2 = (-b-Math.sqrt(d))/(2\*this.a);*

*result = "Корни уравнения: " + this.x1+" "+this.x2;*

*}*

*return result;*

*}*

*public String kv\_voz\_yb()*

*{*

*String result;*

*result = "Экстремум функции находится в точке - " + -this.b/this.a+"\n";*

*result += "Функция возрастает на промежутки от минус бесконечности до "+ -this.b/this.a+ " и убывает начиная с " + -this.b/this.a + " до + бесконечнгсти";*

*return result;*

*}*

*}*

*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 BoolMatrix() {*

*Random r = new Random();*

*this.n = r.nextInt(10)+1;*

*this.m = r.nextInt(10)+1;*

*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 матрицы содержиться %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();*

*}*

*}*

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

*System.out.println("Вариант 1 №9");*

*kv\_yr n = new kv\_yr(1,3,-4);*

*n.kv\_print();*

*System.out.println(n.kv\_korni());*

*System.out.println(n.kv\_voz\_yb());*

*int m=2;*

*kv\_yr[] mass = new kv\_yr[m];*

*Random r = new Random();*

*double max=0, min=9999999;*

*for (int i=0; i!=m; i++)*

*{*

*mass[i] = new kv\_yr(r.nextInt(20)+1,r.nextInt(20)+1,r.nextInt(20)-21);*

*//mass[i].kv\_print();*

*System.out.println(mass[i].kv\_korni());*

*if (mass[i].x1>max && mass[i].x1!=666)*

*{*

*max = mass[i].x1;*

*}*

*if (mass[i].x1<min && mass[i].x1!=666)*

*{*

*min = mass[i].x1;*

*}*

*}*

*System.out.println("Максимальный корень x1 = "+max+". Минимальный корень x1 = "+min);*

*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);*

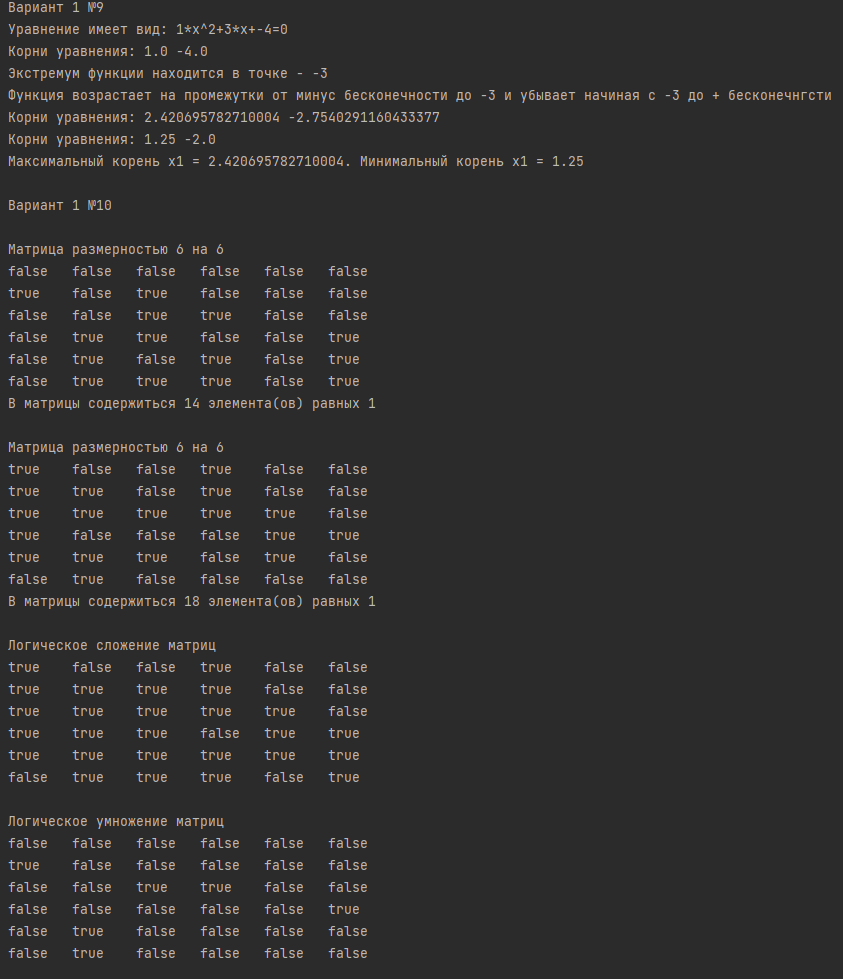
*m\_1.m\_invers();*

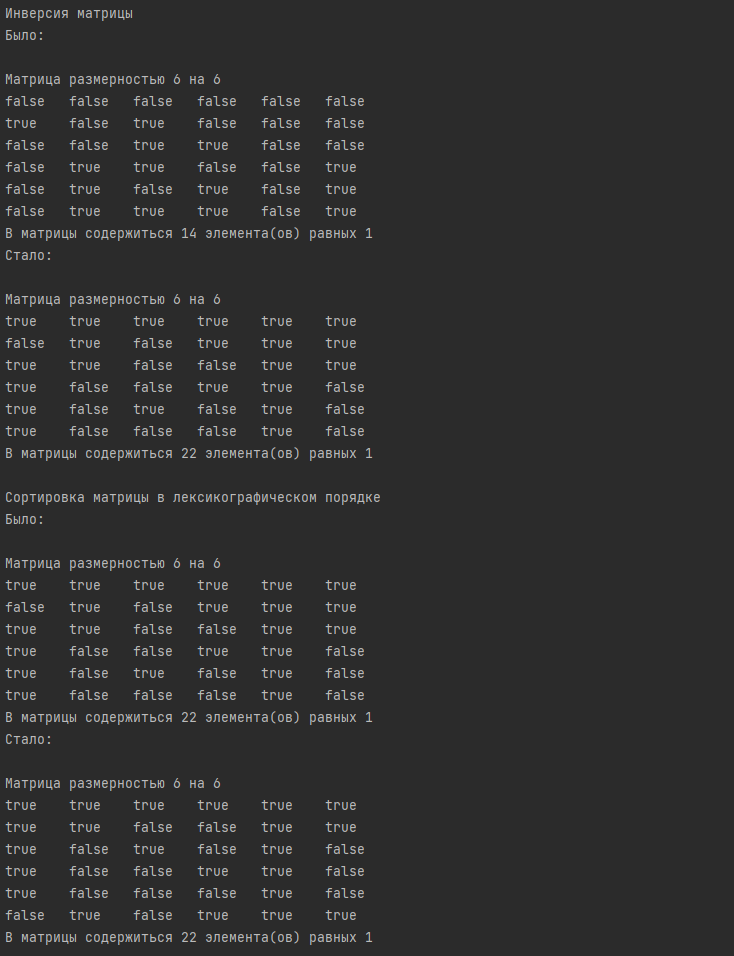
*m\_1.b\_sort();*

*}*

*}*

**Результат выполнения программы:**

****



**Вариант 2, номера 9 и 10.**

9. Product: id, Наименование, UPC, Производитель, Цена, Срок хранения, Количество. Создать массив объектов. Вывести: a) список товаров для заданного наименования; b) список товаров для заданного наименования, цена которых не превосходит заданную; c) список товаров, срок хранения которых больше заданного.

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

**Код программы:**

*package daniil.lab;*

*import java.util.Scanner;*

*import java.util.Random;*

*public class lab3\_2 {*

*public static class Product {*

*//Наименование, UPC, Производитель, Цена, Срок хранения, Количество*

*String name, creator;*

*boolean upc;*

*double cost;*

*int safe\_day, count;*

*public Product(String name, String creator, boolean upc, double cost, int safe\_day, int count) {*

*this.name = name;*

*this.creator = creator;*

*this.upc = upc;*

*this.cost = cost;*

*this.safe\_day = safe\_day;*

*this.count = count;*

*}*

*public Product() {*

*System.out.println("Добавление нового продукта в базу");*

*Scanner in = new Scanner(System.in);*

*int n = in.nextInt();*

*System.out.print("Наименование - ");*

*this.name = in.nextLine();*

*System.out.print("Производитель - ");*

*this.creator = in.nextLine();*

*System.out.print("UPS (T|F) - ");*

*this.upc = in.nextBoolean();*

*System.out.print("Цена - ");*

*this.cost = in.nextInt();*

*System.out.print("Срок хранения (суток) - ");*

*this.safe\_day = in.nextInt();*

*System.out.print("Количество - ");*

*this.count = in.nextInt();*

*}*

*public void Product\_print() {*

*System.out.println("Информация о продукте:");*

*System.out.printf("Наименование - %s\nПроизоводитель - %s\nUPS - %b\nЦена - %f\nСрок хранения - %d\n" +*

*"Количество - %d\n", this.name, this.creator, this.upc, this.cost, this.safe\_day, this.count);*

*System.out.println("");*

*}*

*public static Product[] Pr\_mass\_create(int n) {*

*Product[] mass = new Product[n];*

*Random r = new Random();*

*String[] rand\_creator = {"KIA", "HYUNDAI", "AUDI", "BMW", "NISSAN"};*

*String[] rand\_name = {"K5", "SOLARIS", "RS6", "X7", "X-TRAIL"};*

*for (int i = 0; i != n; i++) {*

*mass[i] = new Product(rand\_name[r.nextInt(4)], rand\_creator[r.nextInt(4)], false, r.nextDouble() \* 100, i \* i + 1, i + 10);*

*}*

*return mass;*

*}*

*public static void select(Product[] mass, int n, String name, int cost, int safe\_day) {*

*for (int i = 0; i != n; i++) {*

*if (!name.equals("") && cost == 0) {*

*if (mass[i].name.equals(name)) {*

*mass[i].Product\_print();*

*}*

*}*

*if (!name.equals("") && cost > 0) {*

*if (mass[i].name.equals(name) && mass[i].cost <= cost) {*

*mass[i].Product\_print();*

*}*

*}*

*if (name.equals("") && cost == 0 && safe\_day > 0) {*

*if (mass[i].safe\_day > safe\_day) {*

*mass[i].Product\_print();*

*}*

*}*

*}*

*}*

*}*

*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 = {"Омкс", "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) {*

*int n=8;*

*System.out.println("Вариант 2 №9");*

*Product[] mass;*

*mass = Product.Pr\_mass\_create(n);*

*for (int i=0; i!=n;i++)*

*{*

*mass[i].Product\_print();*

*}*

*System.out.println("Запрос поиска (а) Поиск K5");*

*Product.select(mass, n, "K5",0,0);*

*System.out.println("Запрос поиска (б) Поиск K5 стоимостью меньше 50");*

*Product.select(mass, n, "K5",50,0);*

*System.out.println("Запрос поиска (с) срок хранения больше 10");*

*Product.select(mass, n, "",0,10);*

*System.out.println("\nВариант 2 №10");*

*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);*

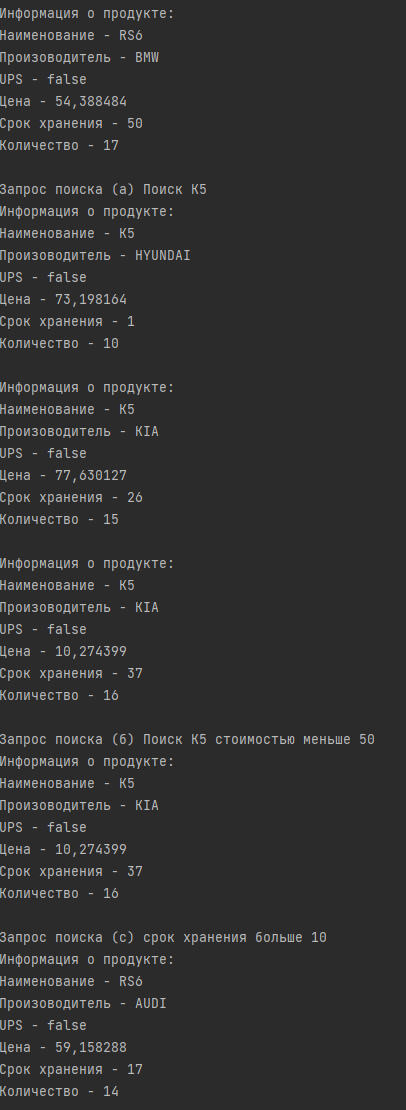
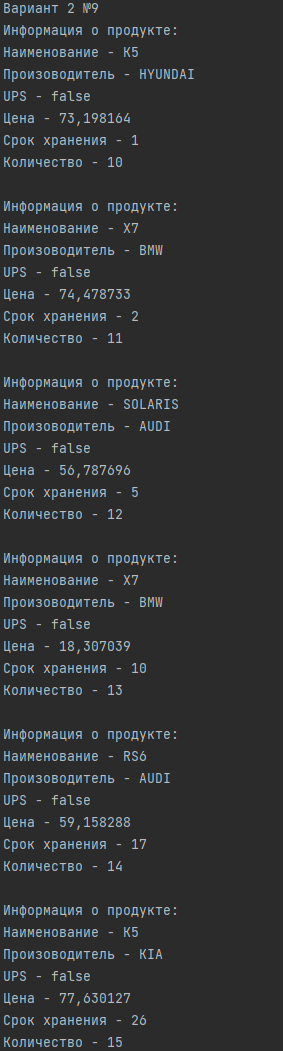
*System.out.println("Запрос (с) список поездов, отправляющихся до заданного пункта назначения и имеющих общие места");*

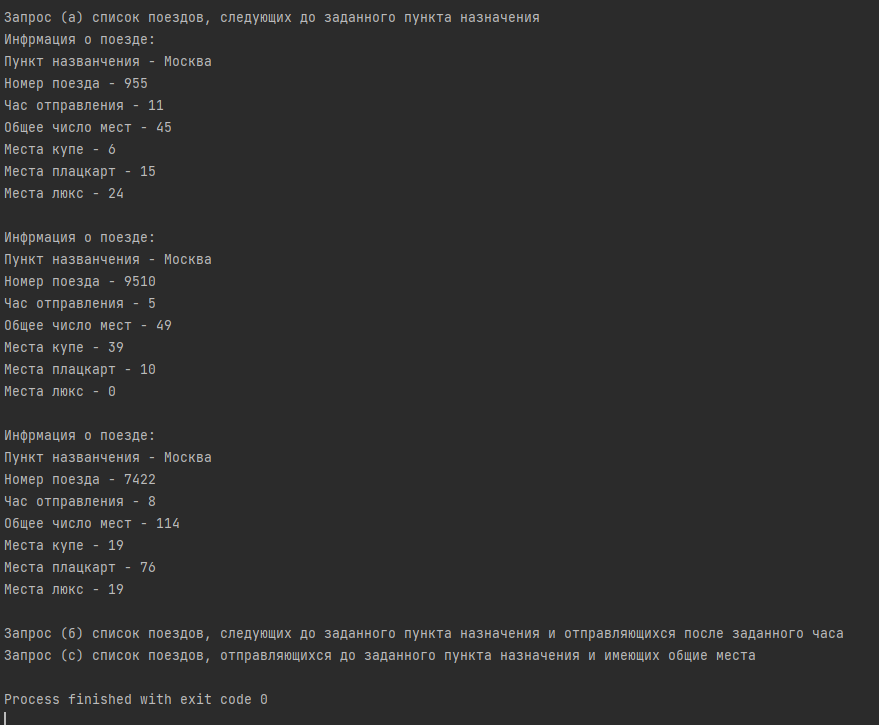
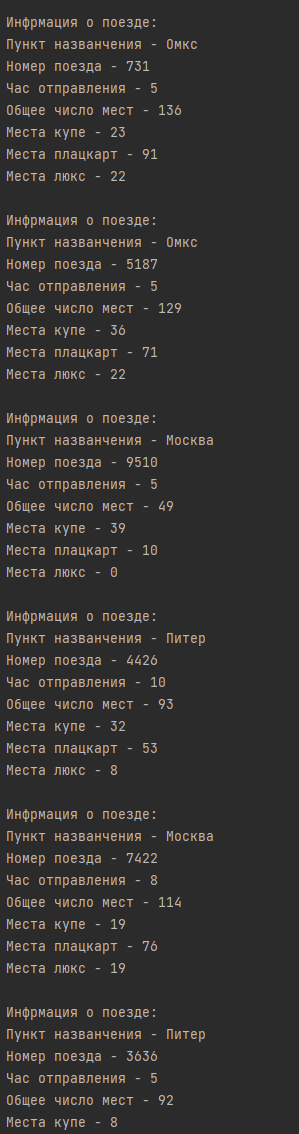
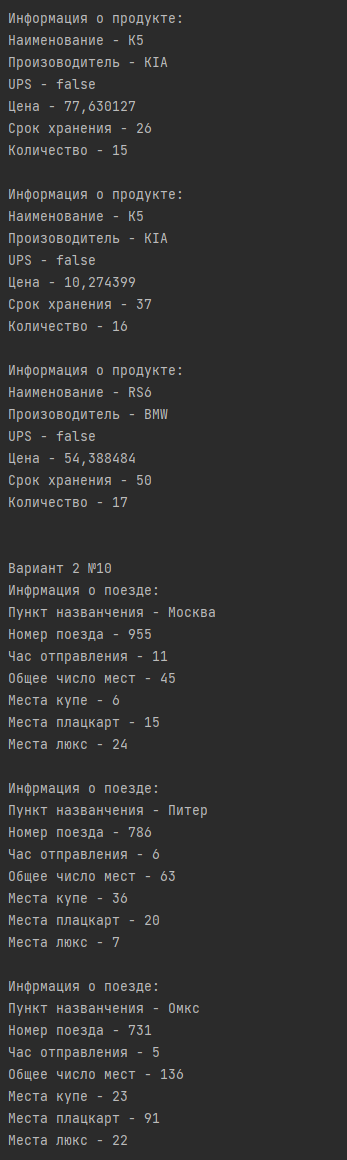
*Train.select(masstr,n,"Надым",0,100);*

*}*

*}*

**Результат выполнения программы:**

****



**Вариант 3, номера 9 и 10.**

9. Создать объект класса Фотоальбом, используя класс Фотография. Методы: задать название фотографии, дополнить фотоальбом фотографией, вывести на консоль количество фотографий.

10. 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.

*Код программы:*

*package daniil.lab;*

*import java.util.\*;*

*import java.time.format.DateTimeFormatter;*

*import java.time.LocalDate;*

*import java.time.Period;*

*public class lab3\_3 {*

*public static class photo*

*{*

*private String ph\_name;*

*private String f\_pass;*

*private photo(String ph\_name, String f\_pass) {*

*this.ph\_name = ph\_name;*

*this.f\_pass = f\_pass;*

*}*

*private photo() {*

*}*

*public boolean equals(photo o2){*

*if (this.hashcode(o2)) {*

*return this.equals(o2);*

*}else{*

*return false;*

*}*

*}*

*public boolean hashcode(photo o2){*

*return this.hashCode() == o2.hashCode();*

*}*

*public String toString(){*

*return "Название фотогравии - "+ this.ph\_name+ ", директория - "+ this.f\_pass;*

*}*

*private void ph\_print(){*

*System.out.println(this.ph\_name+this.f\_pass);*

*}*

*}*

*public static class photoalbum extends photo*

*{*

*private String alb\_name;*

*private ArrayList<photo> album;*

*public photoalbum(String alb\_name, ArrayList<photo> album) {*

*this.alb\_name = alb\_name;*

*this.album = album;*

*}*

*public photoalbum(String alb\_name) {*

*this.alb\_name = alb\_name;*

*this.album = new ArrayList<>();*

*}*

*public boolean equals(photoalbum o2){*

*if (this.hashcode(o2)) {*

*return this.equals(o2);*

*}else{*

*return false;*

*}*

*}*

*public boolean hashcode(photoalbum o2){*

*return this.hashCode() == o2.hashCode();*

*}*

*private void add\_photo(photo new\_p){*

*this.album.add(new\_p);*

*}*

*public String toString(){*

*String result = "";*

*result+="Название альбома - "+ this.alb\_name;*

*for (photo ph:this.album) {*

*result+= "\n" + ph.toString();*

*}*

*return result;*

*}*

*private int ph\_count(){*

*return this.album.size();*

*}*

*}*

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

*System.out.println("Вариант 3 №9");*

*photoalbum album = new photoalbum("Новый альбом");*

*photo ph\_1 = new photo("photo1", "newfolder1");*

*photo ph\_2 = new photo("photo2", "newfolder2");*

*album.add\_photo(ph\_1);*

*album.add\_photo(ph\_2);*

*System.out.println(album.ph\_count());*

*System.out.println(album.toString());*

*System.out.println("\nВариант 3 №10");*

*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("Вевести 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());*

*}*

*public static class day{*

*int day\_number;*

*String notes;*

*public day(int day\_number) {*

*this.day\_number = day\_number;*

*}*

*public day(int day\_number, String notes) {*

*this.day\_number = day\_number;*

*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 day\_number == day.day\_number && Objects.equals(notes, day.notes);*

*}*

*@Override*

*public int hashCode() {*

*return Objects.hash(day\_number, notes);*

*}*

*@Override*

*public String toString() {*

*return "day{" +*

*"day\_number=" + day\_number +*

*", 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, String notes, ArrayList<month> monthes) {*

*this.year\_number = year\_number;*

*this.notes = notes;*

*this.monthes = 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() +*

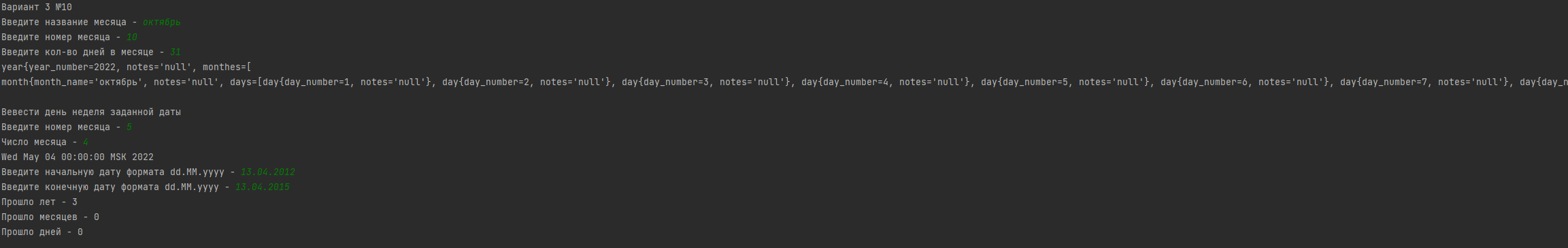
*'}';*

*}*

*}*

*}*

**Результат выполнения программы:**



**Вариант 4, номер 9.**

9. Система Интернет-магазин. Администратор добавляет информацию о Товаре. Клиент делает и оплачивает Заказ на Товары. Администратор регистрирует Продажу и может занести неплательщиков в «черный список».

**Код программы:**

*package dan.lab;*

*import java.io.\*;*

*import java.io.FileOutputStream;*

*import java.util.ArrayList;*

*import java.util.Scanner;*

*public class Lab3\_4 {*

*public static class Item implements Serializable{*

*private static final long serialVersionUID = 2609875448065146411L;*

*private String name;*

*private int cost;*

*private int count;*

*public ArrayList<Item> items\_sell;*

*public Item() {*

*items\_sell = new ArrayList<>();*

*}*

*public Item(String name, int cost, int count) {*

*this.name = name;*

*this.cost = cost;*

*this.count = count;*

*}*

*public void add\_item(String name, int cost, int count){*

*Item cash = new Item(name,cost, count);*

*this.items\_sell.add(cash);*

*}*

*public void add\_count\_item(String name, int cost, int count){*

*int n = this.search\_item(name, cost);*

*if (n!=-1){*

*items\_sell.get(n).count+=count;*

*} else{*

*System.out.println("Такого товара не найдено");*

*}*

*}*

*public void remove\_item(String name, int cost){*

*int n = this.search\_item(name, cost);*

*if (n !=-1) {*

*this.items\_sell.remove(n);*

*}else{*

*System.out.println("Такого товара не найдено");*

*}*

*}*

*public int search\_item(String name, int cost){*

*for (int i=0; i!=items\_sell.size();i++)*

*{*

*if(items\_sell.get(i).name.equals(name) && items\_sell.get(i).cost == cost){*

*return i;*

*}*

*}*

*return -1;*

*}*

*public String toString(int n) {*

*if (n!=-1) {*

*return "item{" +*

*"Название товара='" + this.items\_sell.get(n).name + '\'' +*

*", цена=" + this.items\_sell.get(n).cost + '\'' +*

*", кол-во=" + this.items\_sell.get(n).count +*

*'}';*

*}else{*

*return "Такого товара не найдено";*

*}*

*}*

*@Override*

*public String toString() {*

*return "item{" +*

*"Название товара='" + name + '\'' +*

*", цена=" + cost +*

*", кол-во=" + count +*

*'}';*

*}*

*}*

*public static class Client implements Serializable{*

*private String FIO;*

*private boolean ban;*

*private int balance;*

*public ArrayList<Client> clients;*

*public Client(String FIO, int balance) {*

*this.FIO = FIO;*

*this.ban = false;*

*this.balance = balance;*

*}*

*public Client() {*

*clients = new ArrayList<>();*

*}*

*public void add\_new\_client(String FIO, int balance){*

*Client cash = new Client(FIO, balance);*

*clients.add(cash);*

*}*

*public int search\_client(String name){*

*for(int i=0;i!=this.clients.size();i++){*

*if (this.clients.get(i).FIO.equals(name)){*

*return i;*

*}*

*}*

*return -1;*

*}*

*public void remove\_client(String name)*

*{*

*int n = this.search\_client(name);*

*if (n !=-1){*

*clients.remove(n);*

*}else{*

*System.out.println("такого пользователя не найдено");*

*}*

*}*

*public void add\_money(int n, int money){*

*if (n!=-1){*

*this.balance+=money;*

*}else{*

*System.out.println("такого пользователя не найдено");*

*}*

*}*

*public void ban\_unban\_client(String name, boolean ban){*

*int n = this.search\_client(name);*

*if (n!=-1){*

*clients.get(n).ban = ban;*

*}else{*

*System.out.println("такого пользователя не найдено");*

*}*

*}*

*public String toString(int n) {*

*if (n!=-1) {*

*return "client{" +*

*"ФИО='" + clients.get(n).FIO + '\'' +*

*", забанен=" + clients.get(n).ban +*

*", баланс=" + clients.get(n).balance +*

*'}';*

*}else{*

*return "Такого пользователя не найдено";*

*}*

*}*

*@Override*

*public String toString() {*

*return "client{" +*

*"ФИО='" + FIO + '\'' +*

*", забанен=" + ban +*

*", баланс=" + balance +*

*'}';*

*}*

*}*

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

*System.out.println("Вариант 4 №9");*

*System.out.println("Старт систетмы Интернет-магазин");*

*Item itemss = new Item();*

*itemss.add\_item("IPhone", 6000, 5);*

*FileOutputStream items\_file\_w = new FileOutputStream("itemsobj.txt");*

*ObjectOutputStream itemsobj\_w = new ObjectOutputStream(items\_file\_w);*

*itemsobj\_w.writeObject(itemss);*

*itemsobj\_w.close();*

*Client clientss = new Client();*

*clientss.add\_new\_client("Вася", 2000);*

*FileOutputStream clients\_file\_w = new FileOutputStream("clientsobj.txt");*

*ObjectOutputStream clientsobj\_w = new ObjectOutputStream(clients\_file\_w);*

*clientsobj\_w.writeObject(clientss);*

*clientsobj\_w.close();*

*FileInputStream items\_file = new FileInputStream("itemsobj.txt");*

*ObjectInputStream itemsobj = new ObjectInputStream(items\_file);*

*Object cash\_i = itemsobj.readObject();*

*Item items = (Item) cash\_i;*

*itemsobj.close();*

*System.out.println("Товары загружены");*

*FileInputStream clients\_file = new FileInputStream("clientsobj.txt");*

*ObjectInputStream clientsobj = new ObjectInputStream(clients\_file);*

*Object cash\_c = clientsobj.readObject();*

*Client clients\_list = (Client) cash\_c;*

*clientsobj.close();*

*System.out.println("Клиенты загружены");*

*while (true) {*

*System.out.println("Вы клиент или администратор? 1- клиент, 2 - администратор, 3 - выйти");*

*Scanner in\_1 = new Scanner(System.in);*

*int who = in\_1.nextInt();*

*String cl\_name = "";*

*int cl\_id = 0;*

*while (who != 3 && who != 55 && who != 66) {*

*Scanner in = new Scanner(System.in);*

*switch (who) {*

*case 1:*

*System.out.println("Введите своё имя");*

*cl\_name = (in.nextLine());*

*cl\_id = clients\_list.search\_client(cl\_name);*

*System.out.println("Информация о клиенте:" + cl\_id);*

*System.out.println(clients\_list.toString(cl\_id));*

*if (cl\_id == -1) {*

*who = -1;*

*} else {*

*if (!clients\_list.clients.get(cl\_id).ban) {*

*who = 55;*

*} else {*

*who = -1;*

*System.out.println("Вы были забанены. Обратитесь к администратору");*

*}*

*}*

*break;*

*case 2:*

*System.out.println("Введите пароль");*

*String pass = in.nextLine();*

*if (pass.equals("123")) {*

*System.out.println("Добро пожаловать!");*

*who = 66;*

*} else {*

*System.out.println("Пароль не верный");*

*who = -1;*

*}*

*break;*

*default:*

*System.out.println("Такого значения нет, введите снова");*

*who = -1;*

*break;*

*}*

*if (who == -1) {*

*System.out.println("Вы клиент или администратор? 1- клиент, 2 - администратор, 3 - выйти");*

*who = in.nextInt();*

*}*

*}*

*if (who != 3) {*

*System.out.println("Добро пожаловать в в систему. Доступные команды:");*

*boolean check = true;*

*while (check) {*

*if (who == 55) {*

*System.out.println("Клиент, введите цифры для выполнения действия:\n 1 - Пополнить балланс, 2 - Купить товар, 3 - выйти");*

*Scanner in\_c = new Scanner(System.in);*

*switch (Integer.parseInt(in\_c.nextLine())) {*

*case 1:*

*System.out.println("На какую сумму вы хотите пополнить балланс?" + cl\_name + cl\_id);*

*in\_c.reset();*

*int money = in\_c.nextInt();*

*clients\_list.clients.get(cl\_id).add\_money(cl\_id, money);*

*System.out.println("Успешно");*

*System.out.println(clients\_list.clients.get(cl\_id).toString());*

*break;*

*case 2:*

*System.out.println("Введите название товара и его цену для покупки");*

*in\_c.reset();*

*String it\_name = in\_c.nextLine();*

*in\_c.reset();*

*int it\_cost = in\_c.nextInt();*

*int it\_id = items.search\_item(it\_name, it\_cost);*

*if (it\_id != -1) {*

*if (items.items\_sell.get(it\_id).count > 0) {*

*if (items.items\_sell.get(it\_id).cost <= clients\_list.clients.get(cl\_id).balance) {*

*clients\_list.clients.get(cl\_id).balance -= items.items\_sell.get(it\_id).cost;*

*items.items\_sell.get(it\_id).count -= 1;*

*System.out.println("Покупка совершена успешно");*

*} else {*

*System.out.println("У вас недостаточно средств для покупки");*

*}*

*} else {*

*System.out.println("Товара нет в наличии");*

*}*

*} else{*

*System.out.println("Такого товара нет");*

*}*

*break;*

*case 3:*

*check = false;*

*break;*

*default:*

*System.out.println("Такого значения нет, введите снова");*

*break;*

*}*

*} else {*

*System.out.println("Администратор, введите цифры для выполнения действия:\n 1 - Забанить клиента, 2 - Добавить товар, 3 - Информация о товаре, 4 - Выйти");*

*Scanner in\_a = new Scanner(System.in);*

*switch (Integer.parseInt(in\_a.nextLine())) {*

*case 1:*

*System.out.println("Чтоба занести клиента в чёрный список введите его имя и статус");*

*in\_a.reset();*

*String name = in\_a.nextLine();*

*boolean status = Boolean.parseBoolean(in\_a.nextLine());*

*clients\_list.ban\_unban\_client(name, status);*

*System.out.println("Успешно");*

*break;*

*case 2:*

*System.out.println("Чтобы добавить товар введите его название, цену и кол-во");*

*in\_a.reset();*

*String iname = in\_a.nextLine();*

*int icost = Integer.parseInt(in\_a.nextLine());*

*int icount = Integer.parseInt(in\_a.nextLine());*

*items.add\_item(iname,icost,icount);*

*System.out.println("Успешно");*

*System.out.println(items.toString(items.search\_item(iname,icost)));*

*break;*

*case 3:*

*System.out.println("Введите название товара и его цену");*

*in\_a.reset();*

*String it\_name = in\_a.nextLine();*

*in\_a.reset();*

*int it\_cost = in\_a.nextInt();*

*int it\_id = items.search\_item(it\_name, it\_cost);*

*if (it\_id != -1) {*

*System.out.println(items.items\_sell.get(it\_id).toString());*

*} else {*

*System.out.println("Такого товара нет");*

*}*

*break;*

*case 4:*

*check = false;*

*break;*

*default:*

*break;*

*}*

*}*

*}*

*}*

*//items.add\_item("IPhone", 6000, 5);*

*// FileOutputStream items\_file\_w = new FileOutputStream("itemsobj.txt");*

*// ObjectOutputStream itemsobj\_w = new ObjectOutputStream(items\_file\_w);*

*// itemsobj\_w.writeObject(items);*

*// itemsobj\_w.close();*

*//*

*//*

*// //clients\_list.add\_new\_client("Вася", 2000);*

*// FileOutputStream clients\_file\_w = new FileOutputStream("clientsobj.txt");*

*// ObjectOutputStream clientsobj\_w = new ObjectOutputStream(clients\_file\_w);*

*// clientsobj\_w.writeObject(clients\_list);*

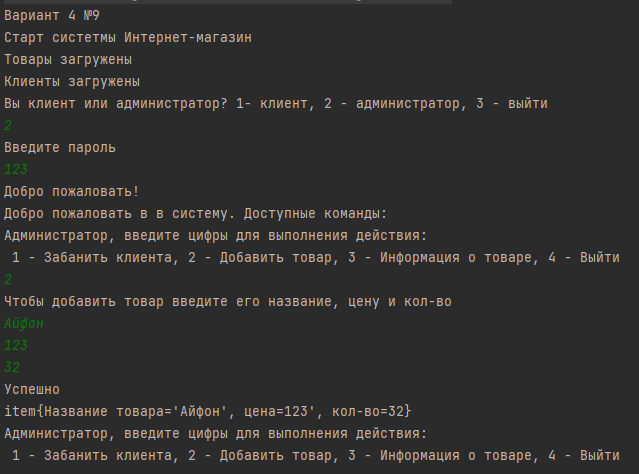
*// clientsobj\_w.close();*

*}*

*}*

*}*

**Результат выполнения программы:**

****

**Вариант 4, номер 10.**

10. 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

**Код программы:**

*package dan.lab;*

*import java.io.\*;*

*import java.util.ArrayList;*

*import java.util.Scanner;*

*public class Lab3\_4\_10 {*

*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");*

*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(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();*

*Passanger passangers = (Passanger) 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 балланс, 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 поплнения");*

*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 - посмотрет 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);*

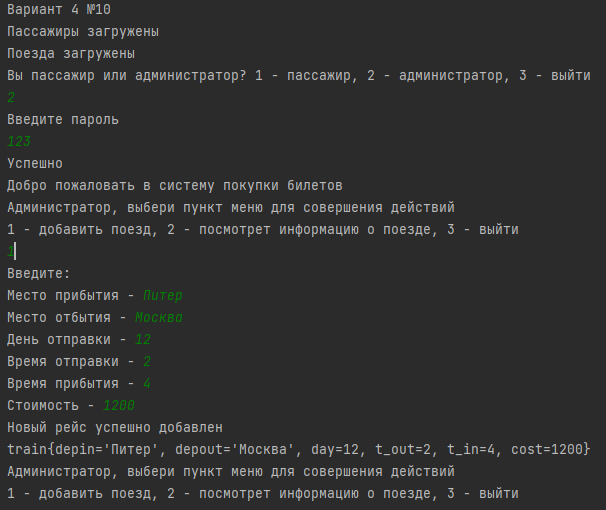
*// trainobj\_w.close();*

*}*

*}*

*}*

**Результат выполнения программы:**



**Вывод:** научились работать с наследованием, полиморфизмом и тд в java.