

МАГИСТЕРСКАЯ ПРОГРАММА 09.04.01/07 Интеллектуальные системы анализа,
обработки и интерпретации больших данных

Вариант 6

Дисциплина: Языки программирования для работы с большими данными

Преподаватель	<hr/>	П.В. Степанов
	(Подпись, дата)	(И.О. Фамилия)

Москва, 2022

Цель работы:

Получение навыков работы с классами Java, исследование механизмов наследования и полиморфизма.

Выполнение:

Задание 1:

1. Определить класс Матрица размерности (m x n). Класс должен содержать несколько конструкторов. Объявить массив объектов. Передать объекты в метод, меняющий местами строки с максимальным и минимальным элементами k-го столбца. Создать метод, который изменяет i-ю матрицу путем возведения ее в квадрат
2. Определить класс Цепная дробь

$$A = a_0 + \frac{x}{a_1 + \frac{x}{a_2 + \frac{x}{a_3 + \dots}}}$$

Определить методы сложения, вычитания, умножения, деления. Вычислить значение для заданного n, x, a[n].

Листинг выполнения подзадачи 1 (файл My_matrix.java)

```
package lr31;

import java.util.ArrayList;

public class My_matrix {
    private int axis_x;
    private int axis_y;
    private int[][] numbers = new int[axis_x][axis_y];
    public My_matrix() {
    }

    public My_matrix(int axis_x, int axis_y, int[][] numbers) {
        this.axis_x = axis_x;
        this.axis_y = axis_y;
        this.numbers = numbers;
    }

    public My_matrix(int axis_x, int axis_y) {
        this.axis_x = axis_x;
        this.axis_y = axis_y;
    }
}
```

```

        this.numbers = new int[axis_x][axis_y];
    }

    public void setAxis_x(int axis_x) {
        this.axis_x = axis_x;
    }

    public void setAxis_y(int axis_y) {
        this.axis_y = axis_y;
    }

    public int getAxis_x() {
        return axis_x;
    }

    public int getAxis_y() {
        return axis_y;
    }

    My_matrix square(){
        My_matrix new_m = new My_matrix(this.axis_x, this.axis_y);
        for (int i = 0; i < this.axis_x; i++) {
            for (int j = 0; j < this.axis_y; j++) {
                int s = 0;
                for (int k = 0; k < this.axis_x; k++) {
                    s += this.numbers[i][k] * this.numbers[k][j];
                }
                new_m.numbers[i][j] = s;
            }
        }
        return new_m;
    }

    public int[][] getNumbers() {
        return numbers;
    }

    public void setNumbers(int[][] numbers) {
        this.numbers = numbers;
    }

    public void setNumber(int i, int j, int k){
        this.numbers[i][j] = k;
    }

    public void change(int k){
        int min = this.numbers[0][k];
        int max = this.numbers[0][k];
        int min_row = 0;
        int max_row = 0;
        for (int i = 0; i < this.axis_x; i++) {
            if (this.numbers[i][k] < min){
                min = this.numbers[i][k];
                min_row = i;
            }
            if (this.numbers[i][k] > max){
                max = this.numbers[i][k];
                max_row = i;
            }
        }
        int[] mid = this.numbers[min_row];
        this.numbers[min_row] = this.numbers[max_row];
        this.numbers[max_row] = mid;
    }
}

```

Листинг выполнения подзадачи 1 (файл lr315.java)

```
package lr31;

import java.util.Arrays;
import java.util.Scanner;
import java.util.concurrent.ThreadLocalRandom;

public class lr315 {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        int amount = in.nextInt();
        My_matrix[] my_array = new My_matrix[amount];
        int n = 3 * amount;
        for (int i = 0; i < amount; i++) {
            int ax_x = ThreadLocalRandom.current().nextInt(2, n + 1);
            int ax_y = ThreadLocalRandom.current().nextInt(2, n + 1);
            my_array[i] = new My_matrix(ax_x, ax_y);
            for (int j = 0; j < ax_x; j++) {
                for (int k = 0; k < ax_y; k++) {
                    my_array[i].setNumber(j, k, ThreadLocalRandom.current().nextInt(-
n, n + 1));
                }
            }
        }
        for (int i = 0; i < amount; i++) {
            System.out.println("Matrix " + i);
            int[][] temp = my_array[i].getNumbers();
            for (int j = 0; j < temp.length; j++) {
                System.out.println(Arrays.toString(temp[j]));
            }
        }
        System.out.println("input k");
        int for_swap = in.nextInt();
        for (int i = 0; i < amount; i++) {
            my_array[i].change(for_swap);
        }
        System.out.println("swapped lines:");
        for (int i = 0; i < amount; i++) {
            System.out.println("Matrix " + i);
            int[][] temp = my_array[i].getNumbers();
            for (int j = 0; j < temp.length; j++) {
                System.out.println(Arrays.toString(temp[j]));
            }
        }
        System.out.println("input i");
        int for_square = in.nextInt();
        my_array[for_square] = my_array[for_square].square();
        System.out.println("Squared matrix");
        for (int i = 0; i < amount; i++) {
            System.out.println("Matrix " + i);
            int[][] temp = my_array[i].getNumbers();
            for (int j = 0; j < temp.length; j++) {
                System.out.println(Arrays.toString(temp[j]));
            }
        }
    }
}
```

```

3
Matrix 0
[1, -4, 7, 9, -8, 8, -8, 6]
[3, 3, -3, -6, 4, 2, 5, 0]
Matrix 1
[-3, -3]
[-3, -6]
[-3, -9]
[-7, 2]
Matrix 2
[8, 3, -3, 8, 1, 2, 3, 7]
[2, 0, -1, 3, 7, -6, 1, 5]
input k
1
swapped lines:
Matrix 0
[3, 3, -3, -6, 4, 2, 5, 0]
[1, -4, 7, 9, -8, 8, -8, 6]
Matrix 1
[-3, -3]
[-3, -6]
[-7, 2]
[-3, -9]
Matrix 2
[2, 0, -1, 3, 7, -6, 1, 5]
[8, 3, -3, 8, 1, 2, 3, 7]

```

```

input i
2
Squared matrix
Matrix 0
[3, 3, -3, -6, 4, 2, 5, 0]
[1, -4, 7, 9, -8, 8, -8, 6]
Matrix 1
[-3, -3]
[-3, -6]
[-7, 2]
[-3, -9]
Matrix 2
[4, 0]
[40, 9]

Process finished with exit code 0

```

Рисунки 1,2 - Результат выполнения кода решения подзадачи 1

Листинг выполнения подзадачи 2 (файл Drob.java)

```
package lr31;

import java.util.ArrayList;
import java.util.Arrays;

public class Drob {
    private int amount;
    private int chisl;
    private int[] znam = new int[this.amount];

    public Drob(int amount, int chisl, int[] znam) {
        this.amount = amount;
        this.chisl = chisl;
        this.znam = znam;
    }

    public Drob() {
    }

    public Drob(int amount, int[] znam) {
        this.amount = amount;
        this.chisl = 1;
        this.znam = znam;
    }

    public int getAmount() {
        return amount;
    }

    public void setAmount(int amount) {
        this.amount = amount;
    }

    public int getChisl() {
        return chisl;
    }

    public void setChisl(int chisl) {
        this.chisl = chisl;
    }

    public int[] getZnam() {
        return znam;
    }

    public void setZnam(int[] znam) {
        this.znam = znam;
    }

    double to_int(){
        double ans = 1;
        for (int i = this.amount - 1; i != 0; i--) {
            ans = this.znam[i] + 1/ans;
        }
        return ans;
    }

    static Drob from_int(double a){
        int[] new_znam = new int[100];
        int i;
        new_znam[0] = (int)a;
        a-=new_znam[0];
        for (i = 1; i < 100; i++) {
            new_znam[i] = (int)(1 / a);
        }
    }
}
```

```

        a = 1/a - new_znam[i];
        if (new_znam[i] == 0){
            break;
        }
    }
    int[] new_new = new int[i];
    for (int j = 0; j < i; j++) {
        new_new[j] = new_znam[j];
    }
    return new Drob(i, 1, new_new);
}

public static Drob sum(Drob a, Drob b){
    return from_int(a.to_int() + b.to_int());
}
public static Drob min(Drob a, Drob b){
    return from_int(a.to_int() - b.to_int());
}
public static Drob mul(Drob a, Drob b){
    return from_int(a.to_int() * b.to_int());
}
public static Drob div(Drob a, Drob b){
    return from_int(a.to_int() / b.to_int());
}

@Override
public String toString() {
    return "Drob{" +
        "amount=" + amount +
        ", chisl=" + chisl +
        ", znam=" + Arrays.toString(znam) +
        '}';
}
}

```

Листинг выполнения подзадачи 2 (файл lr316.java)

```

package lr31;

import java.util.Scanner;
import java.util.concurrent.ThreadLocalRandom;

public class lr316 {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        int amount = in.nextInt();
        Drob[] my_array = new Drob[amount];
        int n = 3 * amount;
        for (int i = 0; i < amount; i++) {
            int a = ThreadLocalRandom.current().nextInt(2, n + 1);
            int b = ThreadLocalRandom.current().nextInt(2, n + 1);
            int[] c = new int[a];
            for (int j = 0; j < a; j++) {
                c[j] = ThreadLocalRandom.current().nextInt(2, n + 1);
            }
            my_array[i] = new Drob(a, b, c);
        }
        System.out.println("my_array:");
        for (int i = 0; i < amount; i++) {
            System.out.println(my_array[i]);
        }
        System.out.println("sum = " + Drob.sum(my_array[0], my_array[1]));
        System.out.println("min = " + Drob.min(my_array[0], my_array[1]));
        System.out.println("mul = " + Drob.mul(my_array[0], my_array[1]));
        System.out.println("div = " + Drob.div(my_array[0], my_array[1]));
    }
}

```

```
}
```

```
4
my_array:
Drob{amount=2, chisl=7, znam=[8, 7]}
Drob{amount=2, chisl=7, znam=[11, 5]}
Drob{amount=9, chisl=6, znam=[5, 2, 11, 2, 4, 12, 2, 9, 6]}
Drob{amount=8, chisl=5, znam=[6, 8, 10, 9, 2, 4, 6, 12]}
sum = Drob{amount=2, chisl=1, znam=[14, 2147483647]}
min = Drob{amount=2, chisl=1, znam=[2, 2147483647]}
mul = Drob{amount=2, chisl=1, znam=[48, 2147483647]}
div = Drob{amount=3, chisl=1, znam=[1, 3, 2147483647]}
```

Рисунок 3 - Результат выполнения кода решения подзадачи 2

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. Book: id, Название, Автор(ы), Издательство, Год издания, Количество страниц, Цена, Переплет. Создать массив объектов. Вывести: а) список книг заданного автора; б) список книг, выпущенных заданным издательством; с) список книг, выпущенных после заданного года.
2. House: id, Номер квартиры, Площадь, Этаж, Количество комнат, Улица, Тип здания, Срок эксплуатации. Создать массив объектов. Вывести: а) список квартир, имеющих заданное число комнат; б) список квартир, имеющих заданное число комнат и расположенных на этаже, который находится в заданном промежутке; с) список квартир, имеющих площадь, превосходящую заданную.

Листинг выполнения подзадачи 1 (файл book.java)

```
package lr32;

public class Book {
    private int id;
    private String name;
    private String author;
    private String publisher;
    private int year;
```



```

private int page_amount;
private int cost;
private boolean cover;

public Book() {

}

public Book(int id, String name, String author, String publisher, int year, int
page_amount, int cost, boolean cover) {
    this.id = id;
    this.name = name;
    this.author = author;
    this.publisher = publisher;
    this.year = year;
    this.page_amount = page_amount;
    this.cost = cost;
    this.cover = cover;
}

public int getId() {
    return id;
}

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

public String getName() {
    return name;
}

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

public String getAuthor() {
    return author;
}

public void setAuthor(String author) {
    this.author = author;
}

public String getPublisher() {
    return publisher;
}

public void setPublisher(String publisher) {
    this.publisher = publisher;
}

public int getYear() {
    return year;
}

public void setYear(int year) {
    this.year = year;
}

public int getPage_amount() {
    return page_amount;
}

public void setPage_amount(int page_amount) {

```

```

        this.page_amount = page_amount;
    }

    public int getCost() {
        return cost;
    }

    public void setCost(int cost) {
        this.cost = cost;
    }

    public boolean isCover() {
        return cover;
    }

    public void setCover(boolean cover) {
        this.cover = cover;
    }

    public static Book[] make_list(int amount){
        return new Book[amount];
    }

    @Override
    public String toString() {
        return "Book{" +
            "id=" + id +
            ", name='" + name + '\'' +
            ", author='" + author + '\'' +
            ", publisher='" + publisher + '\'' +
            ", year=" + year +
            ", page_amount=" + page_amount +
            ", cost=" + cost +
            ", cover=" + cover +
            '}';
    }
}

```

Листинг выполнения подзадачи 1 (файл lr325.java)

```

package lr32;

import java.util.Objects;
import java.util.Scanner;

public class lr325 {
    public static void main(String[] args) {
        Book[] my_book = Book.make_list(5);
        my_book[0] = new Book(1, "1", "1", "1", 1, 1, 1, false);
        my_book[1] = new Book(2, "2", "1", "1", 2, 1, 1, false);
        my_book[2] = new Book(3, "3", "2", "1", 3, 1, 1, false);
        my_book[3] = new Book(4, "4", "2", "1", 4, 1, 1, false);
        my_book[4] = new Book(5, "5", "2", "2", 5, 1, 1, false);
        System.out.println("books of author 2:");
        for (int i = 0; i < 5; i++) {
            if (Objects.equals(my_book[i].getAuthor(), "2")){
                System.out.println(my_book[i]);
            }
        }
        System.out.println("books of publisher 1:");
        for (int i = 0; i < 5; i++) {
            if (Objects.equals(my_book[i].getPublisher(), "1")){
                System.out.println(my_book[i]);
            }
        }
    }
}

```

```

    }
}
System.out.println("books of year later 3:");
for (int i = 0; i < 5; i++) {
    if (my_book[i].getYear() > 3){
        System.out.println(my_book[i]);
    }
}
}
}

```

```

C:\Users\stale\.jdk\openjdk-17.0.2\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ
books of author 2:
Book{id=3, name='3', author='2', publisher='1', year=3, page_amount=1, cost=1, cover=false}
Book{id=4, name='4', author='2', publisher='1', year=4, page_amount=1, cost=1, cover=false}
Book{id=5, name='5', author='2', publisher='2', year=5, page_amount=1, cost=1, cover=false}
books of publisher 1:
Book{id=1, name='1', author='1', publisher='1', year=1, page_amount=1, cost=1, cover=false}
Book{id=2, name='2', author='1', publisher='1', year=2, page_amount=1, cost=1, cover=false}
Book{id=3, name='3', author='2', publisher='1', year=3, page_amount=1, cost=1, cover=false}
Book{id=4, name='4', author='2', publisher='1', year=4, page_amount=1, cost=1, cover=false}
books of year later 3:
Book{id=4, name='4', author='2', publisher='1', year=4, page_amount=1, cost=1, cover=false}
Book{id=5, name='5', author='2', publisher='2', year=5, page_amount=1, cost=1, cover=false}

Process finished with exit code 0

```

Рисунок 4 - Результат выполнения кода решения подзадачи 1

Листинг выполнения подзадачи 2 (файл House.java)

```

package lr32;

public class House {
    private int id;
    private int house_number;
    private int square;
    private int floor;
    private int room_amount;
    private String street;
    private boolean type;
    private int availability_till;

    public House(int id, int house_number, int square, int floor, int room_amount,
String street, boolean type, int availability_till) {
        this.id = id;
        this.house_number = house_number;
        this.square = square;
        this.floor = floor;
        this.room_amount = room_amount;
        this.street = street;
        this.type = type;
        this.availability_till = availability_till;
    }

    public House() {

```

```

}

public int getId() {
    return id;
}

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

public int getHouse_number() {
    return house_number;
}

public void setHouse_number(int house_number) {
    this.house_number = house_number;
}

public int getSquare() {
    return square;
}

public void setSquare(int square) {
    this.square = square;
}

public int getFloor() {
    return floor;
}

public void setFloor(int floor) {
    this.floor = floor;
}

public int getRoom_amount() {
    return room_amount;
}

public void setRoom_amount(int room_amount) {
    this.room_amount = room_amount;
}

public String getStreet() {
    return street;
}

public void setStreet(String street) {
    this.street = street;
}

public boolean isType() {
    return type;
}

public void setType(boolean type) {
    this.type = type;
}

public int getAvailability_till() {
    return availability_till;
}

public void setAvailability_till(int availability_till) {
    this.availability_till = availability_till;
}

```

```

    public static House[] make_list(int amount) {
        return new House[amount];
    }

    @Override
    public String toString() {
        return "House{" +
            "id=" + id +
            ", house_number=" + house_number +
            ", square=" + square +
            ", floor=" + floor +
            ", room_amount=" + room_amount +
            ", street='" + street + '\'' +
            ", type=" + type +
            ", availability_till=" + availability_till +
            '}';
    }
}

```

Листинг выполнения подзадачи 2 (файл lr326.java)

```

package lr32;

import java.util.Objects;

public class lr326 {
    public static void main(String[] args) {
        House[] my_house = House.make_list(5);
        my_house[0] = new House(1, 1, 1, 1, 1, "1", false, 1);
        my_house[1] = new House(2, 1, 2, 2, 1, "1", false, 1);
        my_house[2] = new House(3, 1, 3, 1, 2, "1", false, 1);
        my_house[3] = new House(4, 1, 4, 3, 1, "1", false, 1);
        my_house[4] = new House(5, 1, 5, 4, 1, "1", false, 1);
        System.out.println("flats with 1 room:");
        for (int i = 0; i < 5; i++) {
            if (my_house[i].getRoom_amount() == 1){
                System.out.println(my_house[i]);
            }
        }
        System.out.println("flats with 1 room between 2 and 3 floors:");
        for (int i = 0; i < 5; i++) {
            if (my_house[i].getRoom_amount() == 1 && my_house[i].getFloor() >= 2 &&
my_house[i].getFloor() <= 3){
                System.out.println(my_house[i]);
            }
        }
        System.out.println("flats Square > 2");
        for (int i = 0; i < 5; i++) {
            if (my_house[i].getSquare() > 2){
                System.out.println(my_house[i]);
            }
        }
    }
}

```

```

C:\Users\stale\.jdk\openjdk-17.0.2\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Commu
flats with 1 room:
House{id=1, house_number=1, square=1, floor=1, room_amount=1, street='1', type=false, availability_till=1}
House{id=2, house_number=1, square=2, floor=2, room_amount=1, street='1', type=false, availability_till=1}
House{id=4, house_number=1, square=4, floor=3, room_amount=1, street='1', type=false, availability_till=1}
House{id=5, house_number=1, square=5, floor=4, room_amount=1, street='1', type=false, availability_till=1}
flats with 1 room between 2 and 3 floors:
House{id=2, house_number=1, square=2, floor=2, room_amount=1, street='1', type=false, availability_till=1}
House{id=4, house_number=1, square=4, floor=3, room_amount=1, street='1', type=false, availability_till=1}
flats Square > 2
House{id=3, house_number=1, square=3, floor=1, room_amount=2, street='1', type=false, availability_till=1}
House{id=4, house_number=1, square=4, floor=3, room_amount=1, street='1', type=false, availability_till=1}
House{id=5, house_number=1, square=5, floor=4, room_amount=1, street='1', type=false, availability_till=1}

Process finished with exit code 0

```

Рисунок 5 - Результат выполнения кода решения подзадачи 2

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 Роза, используя классы Лепесток, Бутон. Методы: расцвести, завясть, вывести на консоль цвет бутона.
2. Создать объект класса Дерево, используя классы Лист. Методы: зацвести, опадать листьям, покрыться инеем, пожелтеть листьям.

Листинг выполнения подзадачи 1 (файл Rose.java)

```

package lr33;

import java.util.ArrayList;
import java.util.Objects;

public class Rose {
    private ArrayList<Bud> bud_list;
    private ArrayList<Petal> petal_list;

    public Rose() {
        this.bud_list = new ArrayList<Bud>();
        this.petal_list = new ArrayList<Petal>();
    }

    public ArrayList<Bud> getBud_list() {

```

```

        return bud_list;
    }

    public void setBud_list(ArrayList<Bud> bud_list) {
        this.bud_list = bud_list;
    }

    public ArrayList<Petal> getPetal_list() {
        return petal_list;
    }

    public void setPetal_list(ArrayList<Petal> petal_list) {
        this.petal_list = petal_list;
    }

    @Override
    public boolean equals(Object o) {
        if (this == o) return true;
        if (o == null || getClass() != o.getClass()) return false;
        Rose rose = (Rose) o;
        return Objects.equals(bud_list, rose.bud_list) && Objects.equals(petal_list,
rose.petal_list);
    }

    @Override
    public int hashCode() {
        return Objects.hash(bud_list, petal_list);
    }

    @Override
    public String toString() {
        return "Rose{" +
            "bud_list=" + bud_list +
            ", petal_list=" + petal_list +
            '}';
    }

    public void bloom(String color, int id){
        this.bud_list.add(new Bud(color, id));
        this.petal_list.add(new Petal(color, id));
    }

    public void wither(int id){
        for (int i = 0; i < this.petal_list.size(); i++) {
            if (this.petal_list.get(i).getId() == id){
                this.petal_list.remove(i);
                i--;
            }
        }
        for (int i = 0; i < this.bud_list.size(); i++) {
            if (this.bud_list.get(i).getId() == id){
                this.bud_list.remove(i);
                i--;
            }
        }
    }

    public void colour(){
        for (int i = 0; i < this.bud_list.size(); i++) {
            System.out.println(this.bud_list.get(i).getColor());
        }
    }
}

```

Листинг выполнения подзадачи 1 (файл Petal.java)

```

package lr33;

import java.util.Objects;

public class Petal {
    private String color;
    private int id;

    public Petal(String color, int id) {
        this.color = color;
        this.id = id;
    }

    public int getId() {
        return id;
    }

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

    public Petal() {
    }

    public String getColor() {
        return color;
    }

    public void setColor(String color) {
        this.color = color;
    }

    @Override
    public boolean equals(Object o) {
        if (this == o) return true;
        if (o == null || getClass() != o.getClass()) return false;
        Petal petal = (Petal) o;
        return id == petal.id && color.equals(petal.color);
    }

    @Override
    public int hashCode() {
        return Objects.hash(color, id);
    }

    @Override
    public String toString() {
        return "Petal{" +
            "color='" + color + '\'' +
            ", id=" + id +
            '}';
    }
}

```

Листинг выполнения подзадачи 1 (файл Bud.java)

```

package lr33;

import java.util.Objects;

public class Bud {
    private String color;
    private int id;

    public int getId() {
        return id;
    }

```



```

    }

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

    public Bud(String color, int id) {
        this.color = color;
        this.id = id;
    }

    public Bud() {
    }

    public String getColor() {
        return color;
    }

    public void setColor(String color) {
        this.color = color;
    }

    @Override
    public boolean equals(Object o) {
        if (this == o) return true;
        if (o == null || getClass() != o.getClass()) return false;
        Bud bud = (Bud) o;
        return id == bud.id && Objects.equals(color, bud.color);
    }

    @Override
    public int hashCode() {
        return Objects.hash(color, id);
    }

    @Override
    public String toString() {
        return "Bud{" +
            "color='" + color + '\'' +
            ", id=" + id +
            '}';
    }
}

```

Листинг выполнения подзадачи 1 (файл lr336.java)

```

package lr33;

import java.util.Scanner;

public class lr336 {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        int amount = in.nextInt();
        in.nextLine();
        int j = 0;
        Rose rose = new Rose();
        for (int i = 0; i < amount; i++) {
            rose.bloom(in.next(), in.nextInt());
            in.nextLine();
            System.out.println(rose);
        }
        amount = in.nextInt();
        in.nextLine();
    }
}

```

```

        for (int i = 0; i < amount; i++) {
            rose.wither(in.nextInt());
            System.out.println(rose);
        }
        in.close();
    }
}

```

```

C:\Users\stale\jdk\openjdk-17.0.2\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2021.3.2\lib\idea_rt.jar=65490:C:\Program Files\J
3
4
3
Rose{bud_list=[Bud{color='4', id=3}], petal_list=[Petal{color='4', id=3}]}
2
3
Rose{bud_list=[Bud{color='4', id=3}, Bud{color='2', id=3}], petal_list=[Petal{color='4', id=3}, Petal{color='2', id=3}]}
1
2
Rose{bud_list=[Bud{color='4', id=3}, Bud{color='2', id=3}, Bud{color='1', id=2}], petal_list=[Petal{color='4', id=3}, Petal{color='2', id=3}, Petal{color='1', id=2}]}
3
1
Rose{bud_list=[Bud{color='4', id=3}, Bud{color='2', id=3}, Bud{color='1', id=2}], petal_list=[Petal{color='4', id=3}, Petal{color='2', id=3}, Petal{color='1', id=2}]}
2
Rose{bud_list=[Bud{color='4', id=3}, Bud{color='2', id=3}], petal_list=[Petal{color='4', id=3}, Petal{color='2', id=3}]}
3
Rose{bud_list=[], petal_list=[]}

Process finished with exit code 0
|

```

Рисунок 6 - Результат выполнения кода решения подзадачи 1

Листинг выполнения подзадачи 2 (файл Tree.java)

```

package lr33;

import java.util.ArrayList;
import java.util.Objects;

public class Tree {
    private ArrayList<List> list_array;

    public Tree() {
        this.list_array = new ArrayList<List>();
    }

    public ArrayList<List> getList_array() {
        return list_array;
    }

    public void setList_array(ArrayList<List> list_array) {
        this.list_array = list_array;
    }

    @Override
    public boolean equals(Object o) {
        if (this == o) return true;
        if (o == null || getClass() != o.getClass()) return false;
        Tree tree = (Tree) o;
        return Objects.equals(list_array, tree.list_array);
    }

    @Override
    public int hashCode() {
        return Objects.hash(list_array);
    }

    @Override
    public String toString() {
        return "Tree{" +
            "list_array=" + list_array +

```

```

        '}'';
    }

    public void bloom(String color, int id){
        this.list_array.add(new List(id, color));
    }
    public void leaf_fall(){
        this.list_array = new ArrayList<List>();
    }
    public void frost(){
        for (List list : this.list_array) {
            list.setColor("Blue");
        }
    }

    public void fall(){
        for (List list : this.list_array) {
            list.setColor("Yellow");
        }
    }
}

```

Листинг выполнения подзадачи 2 (файл List.java)

```

package lr33;

import java.util.Objects;

public class List {
    private int id;
    private String Color;

    public List(int id, String color) {
        this.id = id;
        Color = color;
    }

    public List() {
    }

    public int getId() {
        return id;
    }

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

    public String getColor() {
        return Color;
    }

    public void setColor(String color) {
        Color = color;
    }

    @Override
    public boolean equals(Object o) {
        if (this == o) return true;
        if (o == null || getClass() != o.getClass()) return false;
        List list = (List) o;
        return id == list.id && Objects.equals(Color, list.Color);
    }

    @Override
    public int hashCode() {

```

```

        return Objects.hash(id, Color);
    }

    @Override
    public String toString() {
        return "List{" +
            "id=" + id +
            ", Color='" + Color + '\'' +
            '}';
    }
}

```

Листинг выполнения подзадачи 2 (файл Lr337.java)

```

package lr33;

import java.util.Scanner;

public class lr337 {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        int amount = in.nextInt();
        in.nextLine();
        int j = 0;
        Tree tree = new Tree();
        for (int i = 0; i < amount; i++) {
            tree.bloom(in.next(), in.nextInt());
            in.nextLine();
            System.out.println(tree);
        }
        tree.frost();
        System.out.println(tree);
        tree.fall();
        System.out.println(tree);
        tree.leaf_fall();
        System.out.println(tree);
        in.close();
    }
}

```

```

C:\Users\stale\.jdk\openjdk-17.0.2\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Co
3
1 2
Tree{list_array=[List{id=2, Color='1'}]}
34
4
Tree{list_array=[List{id=2, Color='1'}, List{id=4, Color='34'}]}
5 6
Tree{list_array=[List{id=2, Color='1'}, List{id=4, Color='34'}, List{id=6, Color='5'}]}
Tree{list_array=[List{id=2, Color='Blue'}, List{id=4, Color='Blue'}, List{id=6, Color='Blue'}]}
Tree{list_array=[List{id=2, Color='Yellow'}, List{id=4, Color='Yellow'}, List{id=6, Color='Yellow'}]}
Tree{list_array=[]}

Process finished with exit code 0

```

Рисунок 7 - Результат выполнения кода решения подзадачи 2

Задание 4:

Построить модель программной системы.

1. Система Библиотека. Читатель оформляет Заказ на Книгу. Система осуществляет поиск в Каталоге. Библиотекарь выдает Читателю Книгу на абонемент или в читальный зал. При невозвращении Книги Читателем он может быть занесен Администратором в «черный список».
2. Система Конструкторское бюро. Заказчик представляет Техническое Задание (ТЗ) на проектирование многоэтажного Дома. Конструктор регистрирует ТЗ, определяет стоимость проектирования и строительства, выставляет Заказчику Счет за проектирование и создает Бригаду Конструкторов для выполнения Проекта.

Листинг выполнения подзадачи 1 (файл Book.java)

```
package lr34;

import java.util.Objects;

public class Book {
    private int id;
    private String name;

    public Book(int id, String name) {
        this.id = id;
        this.name = name;
    }

    public int getId() {
        return id;
    }

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

    public String getName() {
        return name;
    }

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

    @Override
    public String toString() {
        return "Book{" +
            "id=" + id +
            ", name='" + name + '\'' +
            '}';
    }
}
```

```

    }

    @Override
    public boolean equals(Object o) {
        if (this == o) return true;
        if (o == null || getClass() != o.getClass()) return false;
        Book book = (Book) o;
        return id == book.id && Objects.equals(name, book.name);
    }

    @Override
    public int hashCode() {
        return Objects.hash(id, name);
    }
}

```

Листинг выполнения подзадачи 1 (файл Visitor.java)

```

package lr34;

public class Visitor {
    private int id;

    public Visitor(int id) {
        this.id = id;
    }

    public int getId() {
        return id;
    }

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

    @Override
    public String toString() {
        return "Visitor{" +
            "id=" + id +
            '}';
    }
}

```

Листинг выполнения подзадачи 1 (файл MainForBank.java)

```

package lr34;

import org.jetbrains.annotations.NotNull;

import java.util.HashMap;
import java.util.HashSet;
import java.util.Set;

public class Library {
    private HashMap<Book, Boolean> catalogue;
    private HashMap<Book, Visitor> subscription;
    private HashMap<Book, Visitor> hall;
    private Set<Visitor> black_list;

    public Library() {
        catalogue = new HashMap<>();
        subscription = new HashMap<>();
        hall = new HashMap<>();
        black_list = new HashSet<>();
    }
}

```

```

public void add_book(Book book) {
    catalogue.put(book, true);
}

public void give_book(Visitor visitor, Book book, @NotNull Boolean where) {
    if (catalogue.containsKey(book) && catalogue.get(book)) {
        catalogue.put(book, false);
        if (where) {
            subscription.put(book, visitor);
        } else {
            hall.put(book, visitor);
        }
    }
}

public void return_book(Book book) {
    if (catalogue.containsKey(book)) {
        catalogue.put(book, false);
        subscription.remove(book);
        hall.remove(book);
    }
}

public void add_to_black_list(Visitor visitor) {
    black_list.add(visitor);
}

@Override
public String toString() {
    return "Library{" +
        "catalogue=" + catalogue +
        ", subscription=" + subscription +
        ", hall=" + hall +
        ", black_list=" + black_list +
        '}';
}
}

```

Листинг выполнения подзадачи 1 (файл lr345.java)

```

package lr34;

public class lr345 {
    public static void main(String[] args) {
        Library library = new Library();
        library.add_book(new Book(1, "11"));
        library.add_book(new Book(2, "11"));
        library.add_book(new Book(3, "12"));
        System.out.println(library);
        Visitor v1 = new Visitor(1);
        Visitor v2 = new Visitor(2);
        library.give_book(v1, new Book(1, "11"), true);
        library.give_book(v1, new Book(2, "11"), false);
        library.give_book(v2, new Book(1, "11"), true);
        System.out.println(library);
        library.return_book(new Book(1, "11"));
        library.return_book(new Book(4, "11"));
        System.out.println(library);
        library.add_to_black_list(v1);
        System.out.println(library);
    }
}

```

```

C:\Users\stale\jdk\openjdk-17.0.2\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2021.3.2\lib\idea_rt.jar=49196:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2021.3.2\
Library(catalogue={Book{id=1, name='11'}=true, Book{id=2, name='11'}=true, Book{id=3, name='12'}=true}, subscription={}, hall={}, black_list={})
Library(catalogue={Book{id=1, name='11'}=false, Book{id=2, name='11'}=false, Book{id=3, name='12'}=true}, subscription={Book{id=1, name='11'}=Visitor{id=1}}, hall={Book{id=2, name='11'}=Visitor{id=1}}, black_list={})
Library(catalogue={Book{id=1, name='11'}=false, Book{id=2, name='11'}=false, Book{id=3, name='12'}=true}, subscription={}, hall={Book{id=2, name='11'}=Visitor{id=1}}, black_list={})
Library(catalogue={Book{id=1, name='11'}=false, Book{id=2, name='11'}=false, Book{id=3, name='12'}=true}, subscription={}, hall={Book{id=2, name='11'}=Visitor{id=1}}, black_list={Visitor{id=1}})

Process finished with exit code 0

```

Рисунок 8 - Результат выполнения кода решения подзадачи 1

Листинг выполнения подзадачи 2 (файл Constructor.java)

```

package lr34;

import java.util.Objects;

public class Constructor {
    private final int id;

    public Constructor(int id) {
        this.id = id;
    }

    public int getId() {
        return id;
    }

    @Override
    public String toString() {
        return "Constructor{" +
            "id=" + id +
            '}';
    }

    @Override
    public boolean equals(Object o) {
        if (this == o) return true;
        if (o == null || getClass() != o.getClass()) return false;
        Constructor that = (Constructor) o;
        return id == that.id;
    }

    @Override
    public int hashCode() {
        return Objects.hash(id);
    }
}

```

Листинг выполнения подзадачи 2 (файл Customer.java)

```

package lr34;

import java.util.Objects;

public class Customer {
    private final int id;

    public Customer(int id) {
        this.id = id;
    }

    public int getId() {
        return id;
    }

    @Override
    public String toString() {
        return "Customer{" +
            "id=" + id +

```



```

        '}'';
    }

    @Override
    public boolean equals(Object o) {
        if (this == o) return true;
        if (o == null || getClass() != o.getClass()) return false;
        Customer customer = (Customer) o;
        return id == customer.id;
    }

    @Override
    public int hashCode() {
        return Objects.hash(id);
    }
}

```

Листинг выполнения подзадачи 2 (файл Bureau.java)

```

package lr34;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.Map;
import java.util.concurrent.ThreadLocalRandom;

public class Bureau {
    private HashMap<Customer, Integer> tasks;
    private HashMap<Integer, Integer> cost;
    private HashMap<Constructor, Boolean> team;
    private HashMap<Integer, ArrayList<Constructor>> work_team;

    public Bureau() {
        tasks = new HashMap<>();
        cost = new HashMap<>();
        team = new HashMap<>();
        work_team = new HashMap<>();
    }

    public void add_task(Customer c, int i){
        tasks.put(c, i);
        cost.put(i, ThreadLocalRandom.current().nextInt(10, 1000));
    }

    public void add_to_team(Constructor c){
        team.put(c, true);
    }

    public void create_team(int i){
        work_team.put(i, new ArrayList<>());
        int n = 0;
        for (Map.Entry entry: team.entrySet()){
            if( (Boolean) entry.getValue() ) {
                team.put((Constructor) entry.getKey(), false);
                work_team.get(i).add((Constructor) entry.getKey());
                n++;
                if (n == 2){
                    break;
                }
            }
        }
    }

    public void delete_team(int i){

```

```

        for (int j = 0; j < work_team.get(i).size(); j++) {
            team.put(work_team.get(i).get(j), true);
        }
        work_team.remove(i);
    }

    @Override
    public String toString() {
        return "Bureau{" +
            "tasks=" + tasks +
            ", cost=" + cost +
            ", team=" + team +
            ", work_team=" + work_team +
            '}';
    }
}

```

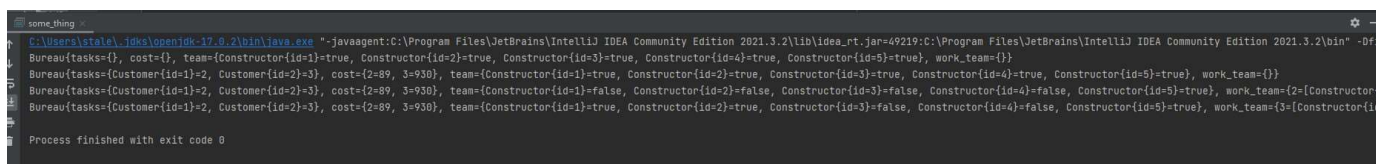
Листинг выполнения подзадачи 2 (файл lr346.java)

```

package lr34;

public class lr346 {
    public static void main(String[] args) {
        Bureau b = new Bureau();
        b.add_to_team(new Constructor(1));
        b.add_to_team(new Constructor(2));
        b.add_to_team(new Constructor(3));
        b.add_to_team(new Constructor(4));
        b.add_to_team(new Constructor(5));
        System.out.println(b);
        b.add_task(new Customer(1), 2);
        b.add_task(new Customer(2), 3);
        System.out.println(b);
        b.create_team(2);
        b.create_team(3);
        System.out.println(b);
        b.delete_team(2);
        System.out.println(b);
    }
}

```



```

C:\Users\stele\Idea\openjdk-17.0.2\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2021.3.2\lib\idea_rt.jar=49219:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2021.3.2\bin" -Df
Bureau{tasks={}, cost={}, team={Constructor{id=1}=true, Constructor{id=2}=true, Constructor{id=3}=true, Constructor{id=4}=true, Constructor{id=5}=true}, work_team={}}
Bureau{tasks={Customer{id=1}=2, Customer{id=2}=3}, cost={2=89, 3=930}, team={Constructor{id=1}=true, Constructor{id=2}=true, Constructor{id=3}=true, Constructor{id=4}=true, Constructor{id=5}=true}, work_team={}}
Bureau{tasks={Customer{id=1}=2, Customer{id=2}=3}, cost={2=89, 3=930}, team={Constructor{id=1}=false, Constructor{id=2}=false, Constructor{id=3}=false, Constructor{id=4}=false, Constructor{id=5}=true}, work_team={2=[Constructor
Bureau{tasks={Customer{id=1}=2, Customer{id=2}=3}, cost={2=89, 3=930}, team={Constructor{id=1}=true, Constructor{id=2}=true, Constructor{id=3}=false, Constructor{id=4}=false, Constructor{id=5}=true}, work_team={3=[Constructor{id
Process finished with exit code 0

```

Рисунок 9 - Результат выполнения кода решения подзадачи 2

Вывод:

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