Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

**(ФГБОУ ВО «КубГУ»)**

**Факультет компьютерных технологий и прикладной математики**

**Кафедра вычислительных технологий**

**ЛАБОРАТОРНАЯ РАБОТА №2**

**Дисциплина: Платформо-независимое программирование**

Работу выполнил: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.С. Агаджанян

Направление подготовки: 02.03.02 Фундаментальная информатика и информационные технологии

Преподаватель: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_В. И. Шиян

**Тема**: РАЗРАБОТКА ПРОГРАММЫ В СРЕДЕ NETBEANS, ОСНОВНЫЕ КОНЦЕПЦИИ ООП, ПРОСТЕЙШИЕ UML-ДИАГРАММЫ.

**Цель**: освоить основы работы со средой разработки NetBeans. Изучить структуру проекта. Освоить процесс построения иерархии классов на основе разработанных UML-диаграмм. Изучить синтаксис и возможности переопределенных (overriding) функций. Разработать программу согласно варианту.

**Задание**:

1. Проверьте пример 3.3. без переопределенного метода equals(), сделайте выводы.

2. Разработайте заданную иерархию классов согласно приведенным ниже вариантам (вариант выдает преподаватель), обязательным элементом является разработка UML-диаграмм классов. Образец - вариант 1 задания. Классы должны быть разложены по пакетам, все классы должны иметь необходимые методы get и set.

3. При разработке конструкторов используйте обращение к суперклассу (super).

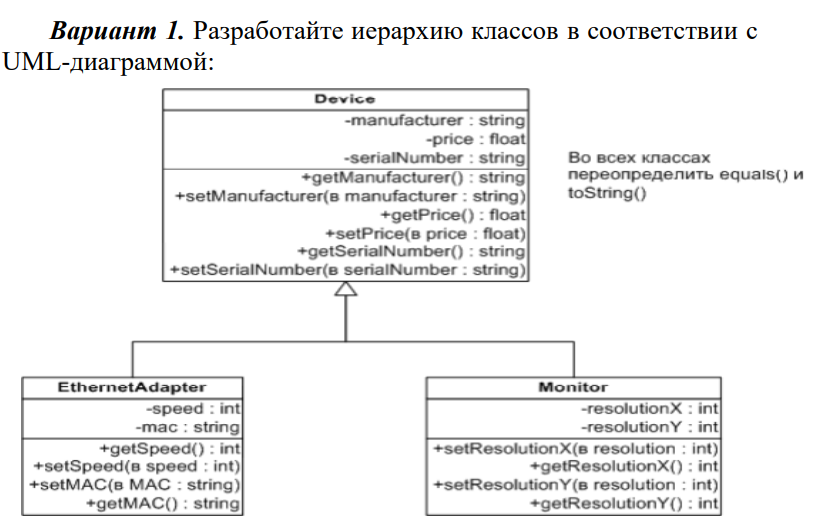
4. В классе main Вашего проекта продемонстрируйте создание

массива разнотипных объектов (подобно примеру 3.2.), установку и вывод характеристик каждого объекта. В массив вносить только уникальные объекты (для проверки уникальности объекта используйте метод equals()).

5. Для вывода содержимого объектов выполните перегрузку метода toString(), начиная с класса-родителя.

6. Методы get и set должны быть определены в каждом неабстрактном классе.

**Вариант 1.**

****

**Проект:**

Изображение выглядит как текст, снимок экрана, Шрифт, линия

Контент, сгенерированный ИИ, может содержать ошибки.

**Device.java**

import java.util.Objects;

public class Device {

    private String manufacturer;

    private float price;

    private String serialNumber;

    public Device(String manufacturer, float price, String serialNumber) {

        this.manufacturer = manufacturer;

        this.price = price;

        this.serialNumber = serialNumber;

    }

    public String getManufacturer() {

        return manufacturer;

    }

    public void setManufacturer(String manufacturer) {

        this.manufacturer = manufacturer;

    }

    public float getPrice() {

        return price;

    }

    public void setPrice(float price) {

        this.price = price;

    }

    public String getSerialNumber() {

        return serialNumber;

    }

    public void setSerialNumber(String serialNumber) {

        this.serialNumber = serialNumber;

    }

    // Базовые методы

    public void replace() {

        System.out.println("Устройство заменено.");

    }

    public void recognize() {

        System.out.println("Устройство распознано.");

    }

    public void installDriver() {

        System.out.println("Драйвер установлен.");

    }

    public void removeDriver() {

        System.out.println("Драйвер удален.");

    }

    @Override

    public boolean equals(Object o) {

        if (this == o) return true;

        if (o == null || getClass() != o.getClass()) return false;

        Device device = (Device) o;

        return Float.compare(device.price, price) == 0 &&

               Objects.equals(manufacturer, device.manufacturer) &&

               Objects.equals(serialNumber, device.serialNumber);

    }

    @Override

    public int hashCode() {

        return Objects.hash(manufacturer, price, serialNumber);

    }

    @Override

    public String toString() {

        return "Device{" +

               "manufacturer='" + manufacturer + '\'' +

               ", price=" + price +

               ", serialNumber='" + serialNumber + '\'' +

               '}';

    }

}

**EthernetAdapter.java**

import java.util.Objects;

public class EthernetAdapter extends Device {

    private int speed;

    private String mac;

    public EthernetAdapter(String manufacturer, float price, String serialNumber, int speed, String mac) {

        super(manufacturer, price, serialNumber);

        this.speed = speed;

        this.mac = mac;

    }

    public int getSpeed() {

        return speed;

    }

    public void setSpeed(int speed) {

        this.speed = speed;

    }

    public String getMac() {

        return mac;

    }

    public void setMac(String mac) {

        this.mac = mac;

    }

    @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;

        EthernetAdapter that = (EthernetAdapter) o;

        return speed == that.speed && Objects.equals(mac, that.mac);

    }

    @Override

    public int hashCode() {

        return Objects.hash(super.hashCode(), speed, mac);

    }

    @Override

    public String toString() {

        return super.toString() + ", EthernetAdapter{" +

               "speed=" + speed +

               ", mac='" + mac + '\'' +

               '}';

    }

}

**Monitor.java**

import java.util.Objects;

public class Monitor extends Device {

    private int resolutionX;

    private int resolutionY;

    public Monitor(String manufacturer, float price, String serialNumber, int resolutionX, int resolutionY) {

        super(manufacturer, price, serialNumber);

        this.resolutionX = resolutionX;

        this.resolutionY = resolutionY;

    }

    public int getResolutionX() {

        return resolutionX;

    }

    public void setResolutionX(int resolutionX) {

        this.resolutionX = resolutionX;

    }

    public int getResolutionY() {

        return resolutionY;

    }

    public void setResolutionY(int resolutionY) {

        this.resolutionY = resolutionY;

    }

    @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;

        Monitor monitor = (Monitor) o;

        return resolutionX == monitor.resolutionX && resolutionY == monitor.resolutionY;

    }

    @Override

    public int hashCode() {

        return Objects.hash(super.hashCode(), resolutionX, resolutionY);

    }

    @Override

    public String toString() {

        return super.toString() + ", Monitor{" +

               "resolutionX=" + resolutionX +

               ", resolutionY=" + resolutionY +

               '}';

    }

}

**Main.java**

import java.util.ArrayList;

import java.util.List;

public class Main {

    public static void main(String[] args) {

        List<Device> devices = new ArrayList<>();

        Device device1 = new Device("Samsung", 100.0f, "12345");

        EthernetAdapter adapter1 = new EthernetAdapter("Intel", 50.0f, "67890", 1000, "00:11:22:33:44:55");

        Monitor monitor1 = new Monitor("LG", 200.0f, "ABCDE", 1920, 1080);

        Device device2 = new Device("Samsung", 100.0f, "12345"); // Дубликат device1

        // Добавление уникальных объектов

        if (!devices.contains(device1)) {

            devices.add(device1);

        }

        if (!devices.contains(adapter1)) {

            devices.add(adapter1);

        }

        if (!devices.contains(monitor1)) {

            devices.add(monitor1);

        }

        if (!devices.contains(device2)) { // Не будет добавлен, так как равен device1

            devices.add(device2);

        }

        // Вывод характеристик каждого объекта

        for (Device device : devices) {

            System.out.println(device);

            // Пример использования методов базового класса

            device.replace();

            device.recognize();

            device.installDriver();

            device.removeDriver();

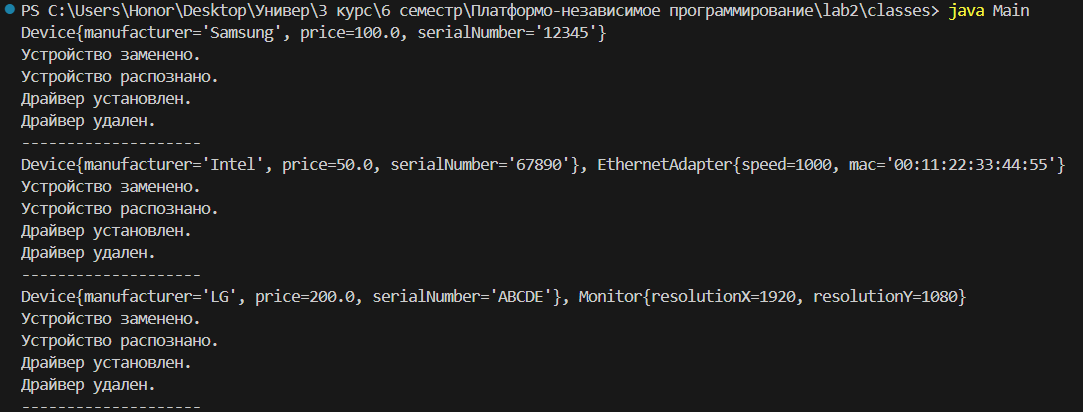
            System.out.println("--------------------");

        }

    }

}

**Результат**

****