МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ
Ордена трудового Красного Знамени федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский технический университет связи и информатики»

Кафедра Математическая кибернетика и информационные технологии

Отчет по Лабораторной №2

Принципы ООП на Java

Выполнил: студент группы

БВТ2402

Козьменко Егор Денисович

Москва, 2025

Ссылка на ГИТ https://github.com/Egor-git-cloud/Lab_works_sem_3-Java-Lab_1.git

Цель работы: Изучение принципов объектно-ориентированного программирования на Java через создание иерархии классов бытовой техники с демонстрацией наследования, инкапсуляции, полиморфизма и абстракции.

Задание 1. Создайте иерархию классов в соответствии с вариантом (11)

Базовый класс: Бытовая техника

Ваша иерархия должна содержать:

- **абстрактный класс;**

HouseholdAppliancesDemo.java > Java > HouseholdAppliancesDemo > main(String[] args)

```java
abstract class HouseholdAppliance {
    protected String brand;
    protected String model;
    protected double power;
    protected boolean isOn;


    protected static int applianceCount = 0;

    public HouseholdAppliance() {
        this.brand = "Unknown";
        this.model = "Unknown";
        this.power = 0.0;
        this.isOn = false;
        applianceCount++;
    }

    public HouseholdAppliance(String brand, String model, double power, boolean isOn) {
        this.brand = brand;
        this.model = model;
        this.power = power;
        this.isOn = isOn;
        applianceCount++;
    }
```

- **два уровня наследуемых классов (классы должны содержать в себе минимум 3 поля и 2 метода, описывающих поведение объекта) +**
- **наличие конструкторов (в том числе по умолчанию);**

1. Класс Refrigerator

```
class Refrigerator extends HouseholdAppliance {
    private double volume;
    private int temperature;
    private boolean hasFreezer;


    public Refrigerator() {
        this.volume = 0.0;
        this.temperature = 4;
        this.hasFreezer = false;
    }

    public Refrigerator(String brand, String model, double power, boolean isOn,
                        double volume, int temperature, boolean hasFreezer) {
        super(brand, model, power, isOn);
        this.volume = volume;
        this.temperature = temperature;
        this.hasFreezer = hasFreezer;
    }

    @Override
    public void performMainFunction() {
        System.out.println("freeze food to a temperature " + temperature + "°C");
    }

    @Override
    public void displayFeatures() {
        System.out.println("Refrigerator Features:");
        System.out.println(" - Volume:" + volume + " l");
        System.out.println("  - Freezer: " + (hasFreezer ? "available" : "not available"));
        System.out.println("  - Current temperature: " + temperature + "°C");
    }
```

2. Класс DishWasher

```
class Dishwasher extends HouseholdAppliance {
    private int countOfShelves;
    private int waterConsumption;
    private String program;

    public Dishwasher() {
        super();
        this.countOfShelves = 0;
        this.waterConsumption = 0;
        this.program = "default";
    }

    public Dishwasher(String brand, String model, double power, boolean isOn,
                      int countOfShelves, int waterConsumption, String program) {
        super(brand, model, power, isOn);
        this.countOfShelves = countOfShelves;
        this.waterConsumption = waterConsumption;
        this.program = program;
    }

    @Override
    public void performMainFunction() {
        System.out.println("Wash dishes according to the program: " + program);
    }

    @Override
    public void displayFeatures() {
        System.out.println("Features of dishwasher:");
        System.out.println("  - Capacity: " + countOfShelves);
        System.out.println("  - Water consumption: " + waterConsumption + " l/cycle");
        System.out.println("  - Current program: " + program);
    }
```

3. Класс SmartRefrigerator

```
class SmartRefrigerator extends Refrigerator {
    private boolean isWifi;
    private double screenSize;

    public SmartRefrigerator() {
        this.isWifi = false;
        this.screenSize = 0.0;
    }

    public SmartRefrigerator(String brand, String model, double power, boolean isOn,
                             double volume, int temperature, boolean hasFreezer, boolean isWifi, double screenSize) {
        super(brand, model, power, isOn, volume, temperature, hasFreezer);
        this.isWifi = isWifi;
        this.screenSize = screenSize;
    }

    @Override
    public void performMainFunction() {
        System.out.println("smartfreeze food to a temperature " + getTemperature() + "°C with AI");
    }

    @Override
    public void displayFeatures() {
        System.out.println("Features of SmartRefrigerator:");
        System.out.println(" - Screen is : " + screenSize + " inch");
        System.out.println("Wi-Fi is " + (isWifi ? "available" : "not available"));
    }
```

- **демонстрацию реализации всех принципов ООП;**

1) Инкапсуляция

```
277         // 2.1 ENCAPSULATION
278         System.out.println("2.1");
279         System.out.println("Initial refrigerator temperature: " + fridge.getTemperature() + "°C");
280         fridge.setTemperature(-5);
281         System.out.println("New refrigerator temperature: " + fridge.getTemperature() + "°C");
282
283         System.out.println("\nInitial dishwasher program: " + dishwasher.getProgram());
284         dishwasher.setProgram("Intensive");
285         System.out.println("New dishwasher program: " + dishwasher.getProgram());
286
287         System.out.println("\nSmart refrigerator Wi-Fi: " + smartFridge.isWifiEnabled());
288         smartFridge.setWifiEnabled(true);
289         System.out.println();
290
```

ПРОБЛЕМЫ 11 ВЫХОДНЫЕ ДАННЫЕ КОНСОЛЬ ОТЛАДКИ ТЕРМИНАЛ ПОРТЫ Run:

```
2. OOP PRINCIPLES DEMONSTRATION:
================================
2.1
Initial refrigerator temperature: 4В°C
New refrigerator temperature: -5В°C

Initial dishwasher program: Auto
Choose program: Intensive
New dishwasher program: Intensive

Smart refrigerator Wi-Fi: false
Wi-Fi is available
```

2) Наследование

```
290
291         // 2.2 INHERITANCE
292         System.out.println("2.2");
293         System.out.println("- Volume: " + smartFridge.getVolume() + " L");
294         System.out.println("- Freezer: " + (smartFridge.hasFreezer() ? "yes" : "no"));
295         System.out.println("- Power consumption: " + smartFridge.getPower() + " kW");
296         System.out.println();
297
```

ПРОБЛЕМЫ 11 ВЫХОДНЫЕ ДАННЫЕ КОНСОЛЬ ОТЛАДКИ ТЕРМИНАЛ ПОРТЫ

```
2.2
- Volume: 600.0 L
- Freezer: yes
- Power consumption: 1.2 kW
```

3) Полиоморфизм

```java
        // 2.3 POLYMORPHISM
        System.out.println("2.3");
        HouseholdAppliance[] appliances = {fridge, dishwasher, smartFridge};

        for (HouseholdAppliance appliance : appliances) {
            System.out.println("\n--- " + appliance.getBrand() + " " + appliance.getModel() + " ---");
            appliance.turnOn();
            appliance.performMainFunction();
            appliance.displayFeatures();
            appliance.turnOff();
        }
        System.out.println();
```

ПРОБЛЕМЫ 11 ВЫХОДНЫЕ ДАННЫЕ КОНСОЛЬ ОТЛАДКИ ТЕРМИНАЛ ПОРТЫ

```
--- LG GC-L247 ---
LG GC-L247 is on
freeze food to a temperature -5B°C
Refrigerator Features:
 - Volume:350.0 l
  - Freezer: not available
  - Current temperature: -5B°C
LG GC-L247 is off

--- Bosch SMS6ZCI00E ---
Bosch SMS6ZCI00E is on
Wash dishes according to the program: Intensive
Features of dishwasher:
  - Capacity: 2
  - Water consumption: 9 l/cycle
  - PЎurrent program: Intensive
Bosch SMS6ZCI00E is off

--- Samsung Family Hub ---
Samsung Family Hub is on
smartfreeze food to a temperature 3B°C with AI
Features of SmartRefrigerator:
```

4) Абстракция

```java
        // 2.4 ABSTRACTION
        System.out.println("2.4");
        System.out.println("All appliances implement abstract methods:");
        fridge.performMainFunction();
        dishwasher.performMainFunction();
        smartFridge.performMainFunction();
        System.out.println();
```

ПРОБЛЕМЫ 11 ВЫХОДНЫЕ ДАННЫЕ КОНСОЛЬ ОТЛАДКИ ТЕРМИНАЛ ПОРТЫ

```
2.4
All appliances implement abstract methods:
freeze food to a temperature -5B°C
Wash dishes according to the program: Intensive
smartfreeze food to a temperature 3B°C with AI
```

- **наличие геттеров и сеттеров;**

```
public String getBrand() {
    return brand;
}
public void setBrand(String brand) {
    this.brand = brand;
}

public String getModel() {
    return model;
}
public void setModel(String model) {
    this.model = model;
}

public double getPower() {
    return power;
}
public void setPower(double power) {
    this.power = power;
}

public boolean isOn() {
    return isOn;
}

public static int getApplianceCount() {
    return applianceCount;
}
```

```
public void setProgram(String program) {
    this.program = program;
    System.out.println("Choose program: " + program);
}


public int getcountOfShelves() {
    return countOfShelves;
}
public void setcountOfShelves(int countOfShelves) {
    this.countOfShelves = countOfShelves;
}

public int getWaterConsumption() {
    return waterConsumption;
}
public void setWaterConsumption(int waterConsumption) {
    this.waterConsumption = waterConsumption;
}

public String getProgram() {
    return program;
}
```

```
public double getVolume() {
    return volume;
}
public void setVolume(double volume) {
    this.volume = volume;
}

public int getTemperature() {
    return temperature;
}

public void setTemperature(int temp) {
    this.temperature = temp;
}

public boolean hasFreezer() {
    return hasFreezer;
}
public void setHasFreezer(boolean hasFreezer) {
    this.hasFreezer = hasFreezer;
}
```

```
public boolean isWifiEnabled() {
    return isWifi;
}

public void setWifiEnabled(boolean isWifi) {
    this.isWifi = isWifi;
    System.out.println("Wi-Fi is " + (isWifi ? "available" : "not available"));
}
```

- **ввод/вывод информации о создаваемых объектах;**

```java
public void info() {
    System.out.println("Brand is: " + brand);
    System.out.println("Model is: " + model);
    System.out.println("Power is: " + power + " kWt");
    System.out.println("Object: " + (isOn ? "is on" : "is off"));
}
```

```java
    @Override
    public void info() {
        super.info();
        System.out.println("Type: Refrigerator");
        System.out.println("Volume: " + volume + " l");
        System.out.println("Current temperature: " + temperature + "°C");
        System.out.println("Freezer: " + (hasFreezer ? "available" : "not available"));
    }
```

```java
    @Override
    public void info() {
        super.info();
        System.out.println("type: Dishwasher");
        System.out.println("Capacity: " + countOfShelves);
        System.out.println("Water consumption: " + waterConsumption + " l/cycle");
        System.out.println("Текущая программа: " + program);
    }
```

```
Complete information about created objects:

--- Refrigerator ---
Brand is: LG
Model is: GC-L247
Power is: 0.85 kWt
Object: is off
Type: Refrigerator
Volume: 350.0 l
Current temperature: -5B°C
Freezer: not available

--- Dishwasher ---
Brand is: Bosch
Model is: SMS6ZCI00E
Power is: 1.1 kWt
Object: is off
type: Dishwasher
Capacity: 2
Water consumption: 9 l/cycle
Current program: Intensive

--- Smart Refrigerator ---
Brand is: Samsung
Model is: Family Hub
Power is: 1.2 kWt
Object: is off
Type: Refrigerator
Volume: 600.0 l
Current temperature: 3B°C
Freezer: available
```

- **предусмотрите в одном из классов создание счетчика созданных объектов с использованием статической переменной, продемонстрируйте работу.**

```
protected static int applianceCount = 0;
```

```
// 5. STATIC VARIABLE - OBJECT COUNTER
System.out.println("5. STATIC VARIABLE - OBJECT COUNTER:");
System.out.println("==================================");

System.out.println("Total household appliances created: " + HouseholdAppliance.getApplianceCount());

// Create more objects for demonstration
Refrigerator extraFridge = new Refrigerator("Indesit", "TIA 40", 0.32, false, 280, 5, false);
Dishwasher extraDishwasher = new Dishwasher("Electrolux", "ESF9452LOX", 0.9, false, 3, 11, "Eco");

System.out.println("After creating additional objects: " + HouseholdAppliance.getApplianceCount());
System.out.println();
```

```
5. STATIC VARIABLE - OBJECT COUNTER:
==================================
Total household appliances created: 6
After creating additional objects: 8
```

Вывод: В ходе лабораторной работы успешно создана трехуровневая иерархия классов, реализующая все принципы ООП. Разработанная система демонстрирует повторное использование кода через наследование, защиту данных через инкапсуляцию, гибкость через полиморфизм и создание абстракций через абстрактные классы и методы. Статический счетчик объектов подтверждает корректность работы конструкторов.