

**МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ СВЯЗИ И МАССОВЫХ  
КОММУНИКАЦИЙ**

**Ордена Трудового Красного Знамени**

**Федеральное государственное бюджетное образовательное учреждение  
высшего образования**

**«Московский технический университет связи и информатики»**

**Кафедра «Математическая Кибернетика и Информационные технологии»**

**Информационные технологии и программирование**

**Лабораторная работа №2**

**Объектно-ориентированное программирование**

**Выполнил: Студент группы**

**БВТ2402**

**Юдин Владимир**

**Москва**

**2025**

## Цель работы

Научиться создавать иерархию классов с использованием принципов ООП: абстракция, инкапсуляция, наследование и полиморфизм. Реализовать конструкторы, геттеры и сеттеры, ввод/вывод информации, а также статический счётчик объектов.

## Задание

1. Создать абстрактный класс.
2. Создать два уровня наследуемых классов, каждый с минимум 3 полями и 2 методами.
3. Реализовать конструкторы (в том числе по умолчанию).
4. Добавить геттеры/сеттеры.
5. Реализовать ввод/вывод информации об объектах.
6. В одном из классов реализовать счётчик созданных объектов.
7. Продемонстрировать работу программы.

## Ход выполнения

- Создан абстрактный класс **HouseholdAppliances**, содержащий общие поля (brand, model, power, isOn) и статическую переменную counter для подсчёта созданных объектов.
- В абстрактном классе определён метод selfTest() (абстрактный) и реализованы общие методы (turnOn, turnOff, printInfo).
- Созданы три класса-наследника:

1. **Refrigerator** — имеет свои поля (temperature, volume, hasFreezer) и методы (defrost, superFreeze).
  2. **DishWasher** — поля (program, volume), методы (changeProgram, переопределение selfTest).
  3. **VacuumCleaner** — поля (isEmpty, cleaningMode), методы (cleanContainer, setCleaningMode).
- В каждом классе реализованы конструкторы по умолчанию и с параметрами, а также геттеры/сеттеры.
  - В методе main создаются объекты разных классов, выводится их информация и демонстрируются принципы ООП.
  - Показана работа статического счётчика объектов: количество увеличивается при каждом создании объекта.

```
2
3 public class ClassPractice {
4     public static void main(String[] args) {
5
6         System.out.println("Создано объектов до: " + HouseholdAppliances.getCounter());
7
8         Refrigerator r = new Refrigerator();
9         DishWasher d = new DishWasher(model:"Bosch", brand:"DW100", power:1800, isOn:true, program:"eco", volume:60);
10        VacuumCleaner v = new VacuumCleaner(brand:"Dyson", model:"V11", power:500, isOn:false, isEmpty:false, cleaningMode:"turbo");
11
12        System.out.println("Создано объектов после: " + HouseholdAppliances.getCounter());
13        System.out.println(x:"-----");
14
15        //холодильник
16        r.printInfo();
17        r.selfTest();
18        r.defrost();
19        r.superFreeze();
20        System.out.println(x:"-----");
21
22        //посудомойка
23        d.printInfo();
24        d.selfTest();
25        d.changeProgram(program:"intensive");
26        System.out.println(x:"-----");
27
28        //пылесос
29        v.printInfo();
30        v.selfTest();
31        v.cleanContainer();
32        v.setCleaningMode(cleaningMode:"silent");
33        System.out.println(x:"-----");
34    }
35 }
```

```
Создано объектов до: 0
Создано объектов после: 3
-----
Бренд: Известный бренд
Модель: Модель известна
Мощность: 0 Вт
Состояние: Включено
Температура: 4С
Объём: 300 л
Морозильная камера: нет
Запуск диагностики холодильника
Датчик температуры неисправен или холодильник выключен
Холодильник уже выключен.
Нельзя сменить режим – холодильник выключен
-----
Бренд: DW100
Модель: Bosch
Мощность: 1800 Вт
Состояние: Включено
Программа: есо
Вместимость: 60 л
Проведена системная диагностика
Программа изменена на intensive
-----
Бренд: Dyson
Модель: V11
Мощность: 500 Вт
Состояние: Включено
Режим уборки: turbo
Контейнер пуст: нет
Контейнер очищен
Режим изменен на auto
Проведена диагностика, очищен контейнер, режим изменен на auto
Контейнер очищен
Режим изменен на silent
-----
```

*Класс Main и вывод консоли*

## Вывод

В ходе работы:

- Реализован абстрактный класс с общими полями и методами.
- Построена иерархия из трёх классов-наследников с двумя уровнями.
- Реализованы конструкторы, геттеры и сеттеры.
- Продемонстрированы все принципы ООП:

- **Абстракция** — абстрактный метод `selfTest()` в базовом классе.
  - **Инкапсуляция** — приватные поля + геттеры/сеттеры.
  - **Наследование** — от `HouseholdAppliances` к `Refrigerator`, `DishWasher`, `VacuumCleaner`.
  - **Полиморфизм** — переопределение метода `selfTest()`.
- Организован ввод/вывод информации об объектах.
  - Продемонстрирована работа статического счётчика созданных объектов.