

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»

Отчёт по лабораторной работе №2

по дисциплине «Информационные технологии и программирование»

Выполнил: студент группы БПИ2404 Журавлев Р. В.

Руководитель: Рыбаков Е. Д.

Москва 2025

Цель работы: изучить основные концепции объектно-ориентированного программирования и применить их на практике на языке программирования Java.

Задание: создайте иерархию классов в соответствии с вариантом. Ваша иерархия должна содержать:

- абстрактный класс;
- два уровня наследуемых классов (классы должны содержать в себе минимум 3 поля и 2 метода, описывающих поведение объекта);
- демонстрацию реализации всех принципов ООП;
- наличие конструкторов (в том числе по умолчанию);
- наличие геттеров и сеттеров;
- ввод/вывод информации о создаваемых объектах;
- предусмотрите в одном из классов создание счетчика созданных объектов с использованием статической переменной, продемонстрируйте работу.

Выбранный вариант: Базовый класс: Бытовая техника. Дочерние классы: Холодильник, Посудомоечная машина, Пылесос.

Labs > Lab2 > HomeAppliance.java > ...

```
1  abstract class HomeAppliance {
2
3      private String brand;
4      private String model;
5      private int power;
6
7      public HomeAppliance() {
8          this.brand = "?";
9          this.model = "?";
10         this.power = 0;
11     }
12
13     public HomeAppliance(String brand, String model, int power) {
14         this.brand = brand;
15         this.model = model;
16         this.power = power;
17     }
18
19     public String getBrand() {
20         return brand;
21     }
22     public void setBrand(String brand) {
23         this.brand = brand;
24     }
25
26     public String getModel() {
27         return model;
28     }
29     public void setModel(String model) {
30         this.model = model;
31     }
32
33     public int getPower() {
34         return power;
35     }
```

```
35     }
36     public void setPower(int power) {
37         this.power = power;
38     }
39
40     public abstract void showInfo();
41     public abstract void function();
42 }
43
```

```
1 public class Dishwasher extends HomeAppliance {
2
3     private int waterConsumption;
4     private int basketsCount;
5     private static int count = 0;
6
7     public Dishwasher() {
8         super();
9         this.waterConsumption = 0;
10        this.basketsCount = 0;
11        count++;
12    }
13
14    public Dishwasher(String brand, String model, int power, int waterConsumption, int basketsCount) {
15        super(brand, model, power);
16        this.waterConsumption = waterConsumption;
17        this.basketsCount = basketsCount;
18        count++;
19    }
20
21    public int getWaterConsumption() {
22        return waterConsumption;
23    }
24    public void setWaterConsumption(int waterConsumption) {
25        this.waterConsumption = waterConsumption;
26    }
27
28    public int getBasketsCount() {
29        return basketsCount;
30    }
31    public void setBasketsCount(int basketsCount) {
32        this.basketsCount = basketsCount;
33    }
34 }
```

```

34
35 @Override
36 public void showInfo() {
37     System.out.println("\nПосудомойка " + getBrand() + " " + getModel() + " - мощность: " + getPower() + "Вт, " + "расход воды: "
38 }
39
40 @Override
41 public void function() {
42     System.out.println("\nПосудомойка " + getBrand() + " " + getModel() + " моет посуду.");
43 }
44
45 public void childrenProtection() {
46     System.out.println("\nУ посудомойки " + getBrand() + " " + getModel() + " включена защита от детей.");
47 }
48 }
49

```

```

Labs > Lab2 > Fridge.java > ...
1 public class Fridge extends HomeAppliance {
2
3     private int temperature;
4     private int capacity;
5     private static int count = 0;
6
7     public Fridge() {
8         super();
9         this.temperature = 0;
10        this.capacity = 0;
11        count++;
12    }
13
14    public Fridge(String brand, String model, int power, int temperature, int capacity) {
15        super(brand, model, power);
16        this.temperature = temperature;
17        this.capacity = capacity;
18        count++;
19    }
20
21    public int getTemperature() {
22        return temperature;
23    }
24    public void setTemperature(int temperature) {
25        this.temperature = temperature;
26    }
27
28    public int getCapacity() {
29        return capacity;
30    }
31    public void setCapacity(int capacity) {
32        this.capacity = capacity;
33    }

```

```

34
35 @Override
36 public void showInfo() {
37     System.out.println("\nХолодильник " + getBrand() + " " + getModel() + " - мощность: " + getPower() + "Вт, " +
38 }
39
40 @Override
41 public void function() {
42     System.out.println("\nХолодильник " + getBrand() + " " + getModel() + " замораживает продукты.");
43 }
44
45 public void leaking() {
46     System.out.println("\nХолодильник " + getBrand() + " " + getModel() + " протёк.");
47 }
48 }
49

```

Labs > Lab2 > VacuumCleaner.java > ...

```
1 public class VacuumCleaner extends HomeAppliance {
2
3     private int capacity;
4     private int noise;
5     private static int count = 0;
6
7     public VacuumCleaner() {
8         super();
9         this.capacity = 0;
10        this.noise = 0;
11        count++;
12    }
13
14    public VacuumCleaner(String brand, String model, int power, int capacity, int noise) {
15        super(brand, model, power);
16        this.capacity = capacity;
17        this.noise = noise;
18        count++;
19    }
20
21    public int getCapacity() {
22        return capacity;
23    }
24    public void setCapacity(int capacity) {
25        this.capacity = capacity;
26    }
27
28    public int getNoise() {
29        return noise;
30    }
31    public void setNoise(int noise) {
32        this.noise = noise;
33    }
```

```
34
35    @Override
36    public void showInfo() {
37        System.out.println("\nПылесос " + getBrand() + " " + getModel() + " - мощность: " + getPower() + "Вт, " + "емкость: " +
38    }
39
40    @Override
41    public void function() {
42        System.out.println("\nПылесос " + getBrand() + " " + getModel() + " чистит помещение.");
43    }
44
45    public void activateTurbo() {
46        System.out.println("\nУ пылесоса " + getBrand() + " " + getModel() + " активирован турбо режим.");
47    }
48 }
49
```

```

Labs > Lab2 > Lab2.java > ...
1  public class Lab2 {
2      Run | Debug
   public static void main(String[] args) {
3
4       Fridge LGfridge = new Fridge(brand:"LG", model:"GA-B509MLS", power:150, -23, capacity:419);
5       Fridge Boschfridge = new Fridge(brand:"Bosch", model:"BD-412M", power:170, -18, capacity:390);
6       VacuumCleaner SamsungCleaner = new VacuumCleaner(brand:"Samsung", model:"SC4581", power:2000, capacity:2, noise:83);
7       Dishwasher EigenDishwasher = new Dishwasher(brand:"Eigen", model:"F451W", power:608, waterConsumption:8, basketsC...3);
8
9       LGfridge.showInfo();
10      Boschfridge.showInfo();
11      LGfridge.function();
12      Boschfridge.leaking();
13
14      SamsungCleaner.showInfo();
15      SamsungCleaner.function();
16      SamsungCleaner.activateTurbo();
17
18      EigenDishwasher.showInfo();
19      EigenDishwasher.function();
20      EigenDishwasher.childrenProtection();
21
22  }
23  }
24

```

```

Холодильник LG GA-B509MLS - мощность: 150Вт, температура: -23°C, объём: 419л, создано холодильников: 2
Холодильник Bosch BD-412M - мощность: 170Вт, температура: -18°C, объём: 390л, создано холодильников: 2
Холодильник LG GA-B509MLS замораживает продукты.
Холодильник Bosch BD-412M протёк.

Пылесос Samsung SC4581 - мощность: 2000Вт, вместимость: 2л, уровень шума: 83дБ, создано пылесосов: 1
Пылесос Samsung SC4581 чистит помещение.
У пылесоса Samsung SC4581 активирован турбо режим.

Посудомойка Eigen F451W - мощность: 608Вт, расход воды: 8л, количество корзин: 3л, создано посудомоек: 1
Посудомойка Eigen F451W моет посуду.
У посудомойки Eigen F451W включена защита от детей.

```

Вывод: на этой лабораторной работе были изучены и реализованы на практике основы объектно-ориентированного программирования на языке программирования Java.