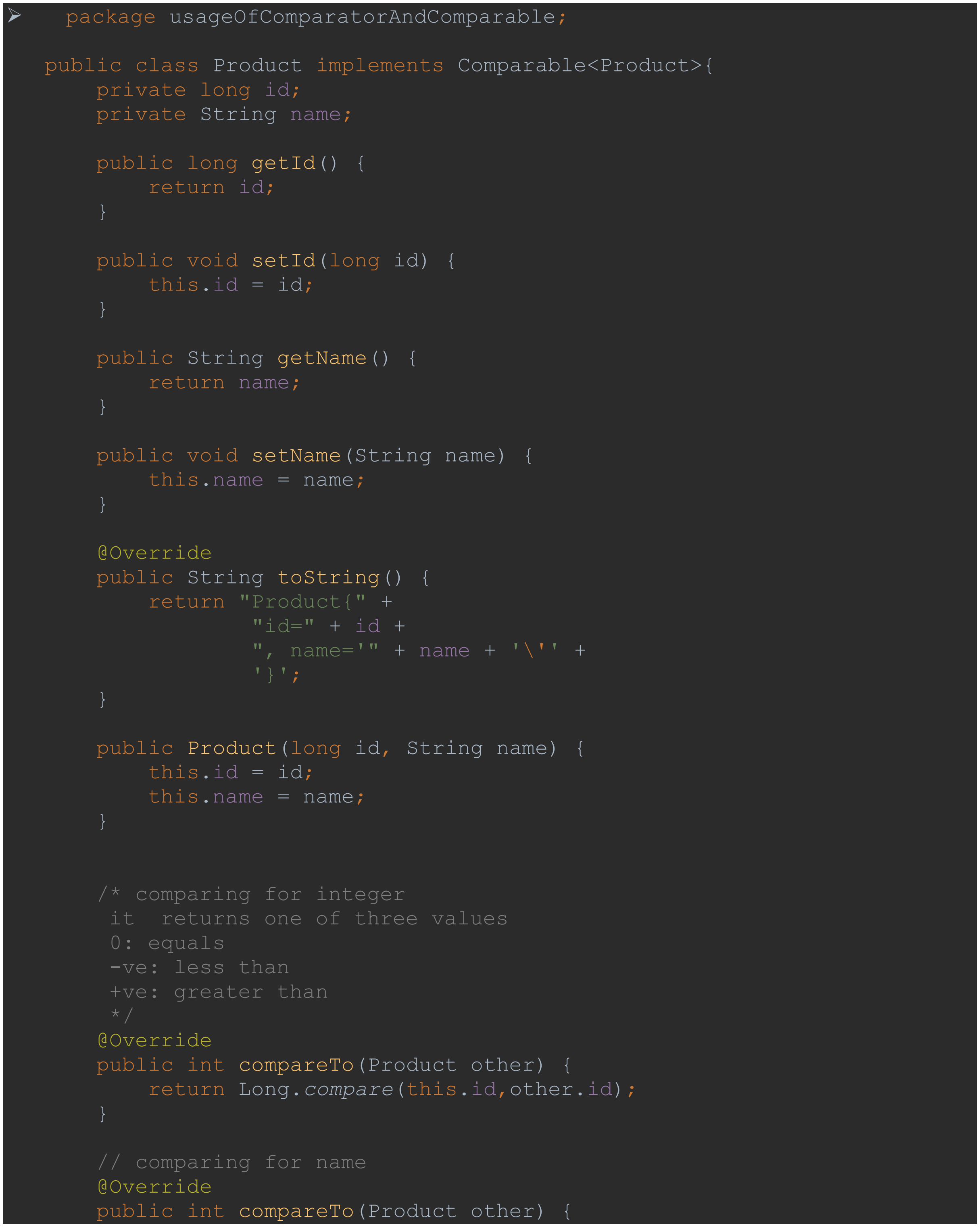
USAGE OF COMPARABALE AND COMPARATOR INTERFACE

- Step 1: for comparable interface with compareTo method



```
➢   package usageOfComparatorAndComparable;

public class Product implements Comparable<Product>{
    private long id;
    private String name;

    public long getId() {
        return id;
    }

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

    public String getName() {
        return name;
    }

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

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

    public Product(long id, String name) {
        this.id = id;
        this.name = name;
    }


    /* comparing for integer
     it  returns one of three values
     0: equals
     -ve: less than
     +ve: greater than
     */
    @Override
    public int compareTo(Product other) {
        return Long.compare(this.id,other.id);
    }

    // comparing for name
    @Override
    public int compareTo(Product other) {
        return this.name.compareTo(other.name);
    }

    // comparing name with ignore case
    @Override
    public int compareTo(Product other) {
        return this.name.compareToIgnoreCase(other.name);
    }



    @Override
    public boolean equals(Object obj) {
        if (this == obj)
            return true;
        if (obj == null || getClass() != obj.getClass())
            return false;
        Product person = (Product) obj;
        return name == person.name;
    }
}
```

- ➢ steps 2: providing implementation of compareTo method and also usage of comparator interface methods

```
➢ package usageOfComparatorAndComparable.impl;

import usageOfComparatorAndComparable.Product;

import java.util.ArrayList;
import java.util.Collections;
import java.util.Comparator;
import java.util.List;

public class ProductImpl {
    public static void main(String[] args) {
        List<Product> personList = new ArrayList<>();
        personList.add(new Product(3, "Alice"));
        personList.add(new Product(1, "Bob"));
        personList.add(new Product(2, "Charlie"));

        // Sorting using Comparable (compareTo method)
        Collections.sort(personList);
        System.out.println("Sorted list using Comparable:");
        for (Product person : personList) {
            System.out.println(person.getId() + ": " + person.getName());
        }

        // Sorting using Comparator (compare method)
//        Comparator<Product> idComparator =
Comparator.comparing(Product::getName);
//        Collections.sort(personList, idComparator);
//        System.out.println("\nSorted list using Comparator:");
//        for (Product person : personList) {
```

```
//            System.out.println(person.getId() + ": " +
person.getName());
//        }

        // Testing equals method
//        Product person1 = new Product(1, "Bob");
//        Product person2 = new Product(1, "Alice");
//        System.out.println("\nTesting equals method:");
//        System.out.println("person1.equals(person2): " +
person1.equals(person2)); // true
    }
    }
```