Task #4 Report

Grocery Store System

Nikoloz Naskidashvili - nikoloz.naskidashvili.1@gmail.com

Overview

Task 4 of this final exam involves the creation of a simple grocery store system. The system is implemented in Java and consists of two main classes: `GroceryStore` and `Product`.

Classes

1. **GroceryStore Class**
   The `GroceryStore` class represents a grocery store entity. It contains the following attributes:
   - `name`: The name of the grocery store.
   - `vatNumber`: The VAT number of the grocery store.
   - `address`: The address of the grocery store.
   - `products`: A list of products that the grocery store sells.
2. The `GroceryStore` class also implements the `LegalEntity` interface, which requires it to provide `getAddress` and `getVatNumber` methods. In addition to these, the `GroceryStore` class provides methods to:
   - Add and delete products from the `products` list.
   - Get the list of products.
   - Save and load the state of the grocery store.
3. The `saveState` and `loadState` methods are used to persist the state of the grocery store by writing the list of products to a file and reading it back.

Java implementation:

```
package finalexam.task4;

import java.util.ArrayList;

import java.util.List;

import java.io.*;

import java.util.ArrayList;
```

```java
import java.util.List;


public class GroceryStore implements LegalEntity {

    public final String name;

    public String vatNumber;

    public String address;

    public final List<Product> products = new ArrayList<>();


    public GroceryStore(String name, String vatNumber, String address) {

        this.name = name;

        this.vatNumber = vatNumber;

        this.address = address;

    }


    @Override

    public String getAddress() {

        return address;

    }


    @Override

    public String getVatNumber() {

        return vatNumber;

    }


    public void addProduct(Product product) {
```

```
        products.add(product);
    }

    public boolean deleteProduct(Product product) {
        for (Product p : products) {
            if (p.equals(product)) {
                products.remove(p);
                return true;
            }
        }

        return false;
    }

    public List<Product> getProducts() {
        return products;
    }

    public void saveState() {
        try {
            BufferedWriter writer = new BufferedWriter(new
FileWriter("task4/products.txt"));
            for (Product p : products) {
                writer.write(p.toString());
                writer.newLine();
            }
```

```
            writer.close();

        } catch (IOException i) {

            i.printStackTrace();

        }

    }



    public void loadState() {

        try {

            BufferedReader reader = new BufferedReader(new
FileReader("task4/products.txt"));

            products.clear();

            String line;

            while ((line = reader.readLine()) != null) {

                String[] parts = line.split(" - ");

                products.add(new Product(parts[0],
Double.parseDouble(parts[1])));

            }

            reader.close();

        } catch (IOException i) {

            i.printStackTrace();

        }

    }

}
```

---

2. **Product Class**
   The `Product` class represents a product that the grocery store sells. It contains the following attributes:
   - `name`: The name of the product.
   - `price`: The price of the product.
3. The `Product` class provides methods to:
   - Get the name and price of the product.
   - Set the price of the product.
   - Convert the product to a string representation.

Java implementation:

```java
package finalexam.task4;


public class Product {

   private final String name;

   private double price;


   public Product(String name, double price) {

       this.name = name;

       this.price = price;

   }


   public String getName() {

       return name;

   }


   public double getPrice() {

       return price;

   }
```

```
    public void setPrice(double price) {
        this.price = price;
    }

    @Override
    public String toString() {
        return name + " - " + price;
    }
}
```

1. The **GrocerStoreTester** class is a driver class that tests the functionality of the GroceryStore and Product classes.

Goals

- **GroceryStore Class**: The goal is to manage the list of products that the grocery store sells and provide methods to manipulate this list.
- **Product Class**: The goal is to represent a product that the grocery store sells and provide methods to manipulate the product's attributes.

Together, these classes form a simple system for managing a grocery store's products.