

OBJECT ORIENTED PROGRAMMING IN JAVA

RESTAURANT MANAGEMENT SYSTEM

29/04/2024

Restaurant Management System

Masterclass in Java

Saba Gelashvili

Chef and Restaurant Consultant

Saba.gelashvili.2@iliauni.edu.ge

Description

Create a Restaurant Management System (RMS) in Java. RMS software is widely used in the food industry. Our example is a basic one with the following features:

1. Storage for menu items
2. Ability to add a new menu item
3. Ability to remove a menu item
4. Ability to print the full menu on the console

RMS Structure

We will need the following classes:

1. MenuItem - Represents a menu item
2. RMS - The restaurant management system
3. RestaurantTester - For testing the management system

Class MenuItem

The MenuItem class should have fields for name, description, and price. It can be implemented like:

```
public class MenuItem {  
    private String name, description;  
    private double price;  
    // Getters and setters  
    // Implement toString()  
}
```

Notice the private fields and the getters/setters. All fields should be private unless requirements state otherwise. Read about toString() and implement it.

Class RMS

The RMS should have an internal data structure to store menu items. It needs methods to add/remove items and print the full menu. Like:

```
public class RMS {  
    private List<MenuItem> menu = new ArrayList<>();  
    public void addMenuItem(MenuItem item) {  
        menu.add(item);  
    }  
    public boolean removeMenuItem(MenuItem item) {  
        // Implementation to remove item  
    }  
    public void printMenu() {  
        // Implementation to print menu  
    }  
}
```

Notice usage of ArrayList, for-loops, String comparison. This demonstrates interfaces and booleans.

Class RestaurantTester

Let's test our system. Create some MenuItems, instantiate RMS, add/remove items, print final menu.

```
public class RestaurantTester {  
    public static main(String[] args) {  
        // Create menu items  
        // Create RMS instance  
        // Add/remove menu items  
        // Print final menu  
    }  
}
```

We print the menu to verify everything works correctly.