

CENG 112 - Data Structures

Assignment 4: Food Delivery Application

In this homework, you are expected to implement a “Food Delivery Application” using Java. This homework will cover the topics given below:

- 1) Generics, Interfaces
- 2) Queue ADT, Stack ADT
- 3) Iterators, Comparators
- 4) Binary Search Trees

Assume that you are designing a food delivery application that runs as a marketplace for local restaurants. The restaurants and foods are placed in several binary search trees with respect to different attributes such as price, stock, restaurant rating, and delivery time.

Implement a ***Restaurant*** class that includes but is not limited to:

```
public class Restaurant {
    private String name
    private double rating
    private String cuisine
    private int deliveryTime    // in minutes
    public void updateCuisine(String category)
    public void updateRating(double score)
    public void updateDeliveryTime(int deliveryTime)
    ... // Constructors, getters, setters and other methods
}
```

Implement the ***Orderable*** interface that includes but is not limited to:

```
public interface Orderable {
    public void updatePrice(double price)
    public void updateStock(int stock)
    ... // Other method names if needed
}
```

Implement a ***Food*** class that includes but is not limited to:

```
public class Food implements Orderable ... // other interfaces if needed
{
    private String name
    private double price
    private int stock
    private Restaurant restaurant
    ... // Constructors, getters, setters and other methods
}
```

You are expected to read restaurant and food data from CENG112_HW4.txt file where each line is formed as:

fName, fPrice, fStock, rName, rRating, rCuisine, rDelivery

Note that the initial letters “f” and “r” represent food and restaurant attributes, respectively.

You should assume that restaurant names are unique and **create only one instance** if there are multiple restaurants with the same name in the text file. In other words, before adding a new restaurant into the BST, check if it already exists.

Perform the following operations step-by-step:

- 1) List the names and ratings of the restaurants in descending order of rating.
- 2) List the names, prices, and stocks of the food in ascending order of price.
- 3) Print the name of the Pizza restaurant that has the shortest delivery time.
- 4) Print the name of the Coffee with the highest amount of stock.
- 5) List and remove the foods (from Food BST) that are more expensive than 80 TRY.
- 6) List and remove the restaurants (from Restaurant BST) that are rated less than 8.0.
- 7) Increase all food prices by 20% (use the updatePrice method).
- 8) Cut all food stocks by half (use the updateStock method).
- 9) List the names and ratings of the restaurants in descending order of rating. (Like Step 1).
- 10) List the names, prices, and stocks of the food in ascending order of price. (Like Step 2).

Assignment Rules

- This is a group assignment (2 students). However, inter-group collaboration is not allowed!
- All assignments are subject to plagiarism detection and the suspected violations (the solutions derived from or inspired by the solution of other groups) cause to be graded as zero.
- It is not allowed to use Java Collections Framework.
- Your code should be easy to read and test:
 - Keep your code clean. Avoid duplication and redundancy. 🔗
 - Follow Java Naming Conventions. 🔗
 - Use relative paths instead of absolute ones. 🔗

Submission Rules

All submissions must:

- be performed via **Microsoft Teams** by only one of the group members,
- be exported as an Eclipse Project and saved in ZIP format,
- include all necessary data files (TXT, CSV, JSON, etc.) in the right directory,
- follow a specific naming convention such that CENG112_HW4_groupID.

Eclipse Project: CENG112_HW4_G5

Exported Archive File: CENG112_HW4_G5.zip

Submissions that do not comply with the rules above are penalized.

Those who want to change groups can send their requests on Microsoft Teams.