

CENG 211 PROGRAMMING FUNDAMENTALS

HOMEWORK-4

Due Date: 03 December 2017, 23:55

You are required to write a Java program for simulating diet application. You are required to decide user's meals according to Breakfast, Lunch, Dinner. The diet application should check whether total calories of chosen meals do not exceed optimum calorie value.

There are two classes of nutrition, as Food and Drink. Nutrition class should have calorie and name of the nutrition whereas the unit of Food is gr. and the unit of Drink is the ml.

***** Each meal has at least the following properties and methods:

- total calorie,
- nutrition(s),
- nutrition type(s),
- computeTotalCalorie().

***** A Breakfast should have the following properties:

- total calorie,
- nutrition(s),
- nutrition type(s) —> cereal, dairy products, fruits, vegetables,
- computeTotalCalorie().

***** A Lunch should have the following properties:

- total calorie,
- nutrition(s),
- nutrition type(s) —> meats, fruits, vegetables,
- computeTotalCalorie().

***** A Dinner should have the following properties:

- total calorie,
- nutrition(s),
- nutrition type(s) —> fish, fruits, vegetables,
- computeTotalCalorie().

***** All nutrition types should be enum —> CEREAL, MEATS, DAIRY-PRODUCTS, FRUITS, FISH, VEGETABLES.

***** You should read nutrition information from file, named “**nutrition.dat**”. (File format: name, unit (ml., gr.), calorie, nutrition type, Food/Drink).

***** You should design graphical user interface (GUI) by using SWING programming. In this GUI, following operations should be available in the given sequence:

1. User should enter his/her weight, height, age and gender. You should compute daily optimum calories that user should take according to formula below.

Female —> $10 * \text{weight (in kg)} + 6.25 * \text{height (in cm)} - 5 * \text{age} - 161$

Male —> $10 * \text{weight (in kg)} + 6.25 * \text{height (in cm)} - 5 * \text{age} + 5$

2. User should be able to choose meal with a **radio button**. Then, corresponding nutrition should be displayed to user in a swing table where user is able to sort ascending/descending order based on calorie values. User can add nutrition using an **add button**.
3. After all nutrition for each meal are selected user should have a chance to check whether he/she exceed his/her daily calorie limit or not using a **check button**.

4. After all nutrition for each meal are selected and checked, user should be able to save daily meal information to file.

***** You should write daily meal information to file, named “dailyNutritions.dat”.

(File format:

meal name - nutrition name, unit (ml., gr.), calorie, nutrition type, Food/Drink;
nutrition name, unit (ml., gr.), calorie, nutrition type, Food/Drink;...

meal name - nutrition name, unit (ml., gr.), calorie, nutrition type, Food/Drink;
nutrition name, unit (ml., gr.), calorie, nutrition type, Food/Drink;...).

<p><input type="radio"/> Breakfast</p> <p><input checked="" type="radio"/> Lunch</p> <p><input type="radio"/> Dinner</p> <p><input type="button" value="Add"/></p>	<table border="1"><thead><tr><th>name</th><th>unit</th><th>cal</th></tr></thead><tbody><tr><td>Apple</td><td>100</td><td>44</td></tr><tr><td>Broccoli</td><td>27</td><td>95</td></tr><tr><td>Spinach</td><td>8</td><td>100</td></tr><tr><td>Banana</td><td>150</td><td>107</td></tr><tr><td>Orangejuice</td><td>200</td><td>150</td></tr><tr><td>Chicken</td><td>220</td><td>200</td></tr><tr><td>Beef</td><td>100</td><td>300</td></tr></tbody></table>	name	unit	cal	Apple	100	44	Broccoli	27	95	Spinach	8	100	Banana	150	107	Orangejuice	200	150	Chicken	220	200	Beef	100	300
name	unit	cal																							
Apple	100	44																							
Broccoli	27	95																							
Spinach	8	100																							
Banana	150	107																							
Orangejuice	200	150																							
Chicken	220	200																							
Beef	100	300																							

Figure 1. An example UI of your application.

IMPORTANT NOTES:

- You should use one of the **layouts** for your GUI.
- You should have at least three different packages. **domain** layer, **data access** layer, **presentation** layer.
- You should use **INHERITANCE** approach for meal relations and nutrition relations (Food and Drink).

SUBMISSION RULES:

- You should create your Java project as **ID1_ID2_HW4** and export as **ID1_ID2_HW4.zip**
- You should upload your zip file **ID1_ID2_HW4.zip** to the CMS.
- One of the group members is sufficient to upload homework to the CMS.
- You should add an author comment to the top of each class that you implement.