LAB 211 Assignment

Type: Long Assignment
Code:
LOC: 500
Slot(s): N/A

Title

Manage ingredients at the coffee store

Background

The coffee store F **sells many types of drinks** that are prepared and combined from many different ingredients.

For example: a strawberry smoothie is combined with strawberries, milk, sugar, ...; milk coffee prepared from coffee and milk; ...

Store F wants to build a tool to manage menus and ingredients that make operations more convenient. The software stores data in **three files**:

1. The **Ingredients.dat file** is used to store information about ingredients.
2. The **Menu.dat file** is used to store recipe information.
3. The **Order.dat file** stores prepared drink information and sold.

Program Specifications

Build a management program. With the following basic functions

1. **Build your data structure**
2. **Manage ingredients**
    1.1. Add an ingredient
    1.2. Update ingredient information.
    1.3. Delete ingredient.
    1.4. Show all ingredients.
3. **Manage beverage recipes**
    2.1. Add the drink to menu.
    2.2. Update the drink information.
    2.3. Delete the drink from menu.
    2.4. Show menu.
4. **Dispensing beverages**
    3.1. Dispensing the drink.
    3.2. Update the dispensing drink.

5. **Report**

    4.1. The ingredients are available.

    4.2. The drinks for which the store is out of ingredients.

    4.3. Show all the dispensing drink.

6. **Store data to files**

7. **Exit.**

Each menu choice should invoke an appropriate function to perform the selected menu item. Your program must display in the menu after each task and wait for the user to select another option until the user chooses to quit the program.

Features:

This system contains the following functions:

Display a menu and ask users to select an option.

1. **Build your data structure (50 LOCs)**

- The ingredients' information: **code, name, and other attributes.**
- The menu information: **code, name, and list of ingredients with quantities.**
- **Classes, abstract classes, interfaces are required** for applying in building application.
- **Student must implement the polymorphism properties** of object-oriented programming.

2. **Manage ingredients (80 LOCs):**

    1.1. Add an ingredient. (20 LOCs)

    - ✓ Create a submenu.
    - ✓ **Remember that the constraints must be checked: code can not duplicate**
    - ✓ **Add the new ingredient to the collection.**
    - ✓ Ask continuously create new ingredients or go back to the main menu.

    1.2. Update ingredient information. (20 LOCs)

    - ✓ **Require to enter the ingredient code**
    - ✓ If a code does not exist, the notification "**The ingredient does not exist**". Otherwise, we can start inputting new information about ingredients and update.
    - ✓ If new information is blank, then not change old information.
    - ✓ Remember new information must be validated.
    - ✓ The system must print out the result of the update.
    - ✓ After updating, the program returns to the main screen.

1.3. Delete ingredient. (20 LOCs)

- ✓ **Before the delete system must show, confirmed message.**
- ✓ **Show the result of to delete: success or fail.**
- ✓ After deleting, the program returns to the main screen.

1.4. Show all the ingredients. (20 LOCs).

- ✓ Show all the data in the ingredients' collection on the screen.
- ✓ **Sorted in ascending order by name**

3. Manage beverage recipes (120 LOCs)

2.1. Add the drink to the menu. (40 LOCs)

- ✓ Create a submenu.
- ✓ **Remember that the constraints must be checked: ingredient code**
- ✓ Add the new ingredient to the list of ingredients.
- ✓ Ask continuously add new ingredients or show a confirmed message.
- ✓ After confirming, the result of the drink: success or fail.
- ✓ After confirming, the program returns to the main screen.

2.2. Update the drink information. (40 LOCs)

- ✓ Create a submenu.
- ✓ **Remember that the constraints must be checked: menu's code, ingredient code.**
- ✓ If the menu's code does not exist, the notification says "**The drink does not exist**". Otherwise, we can start inputting new information about drinks and update.
- ✓ If new information is blank, then not change old information.
- ✓ Remember new information must be validated.
- ✓ The system must print out the result of the update.
- ✓ After updating, the program returns to the main screen.

2.3. Delete the drink from the menu. (20 LOCs)

- ✓ **Before the delete system must show, confirmed message.**
- ✓ **Show the result of to delete: success or fail.**
- ✓ After deleting, the program returns to the main screen.

2.4. Show all the menu. (20 LOCs).

- ✓ Show all the data in the drink collection on the screen.
- ✓ **Sorted in ascending order by name**

4. **Dispensing beverages (100 LOCs)**

4.1. Dispensing the drink. (50 LOCs)

- ✓ The system searches for the drink list.
- ✓ If the menu's code does not exist, the notification says "**The drink does not exist**". Otherwise, we can start **checking ingredients** of the drink and **update the ingredient's status**.
- ✓ Remember new information must be validated.
- ✓ The system must print out the result: success or fail.
- ✓ The program returns to the main screen.

4.2. Update the dispensing drink. (50 LOCs)

- ✓ **Update quantity of the dispensing drink**
- ✓ If the dispensing drink code does not exist, the notification "**The drink does not exist**". Otherwise, we can **update quantity, start checking ingredients for dispensing the drink and update the ingredient's status**.
- ✓ Remember new information must be validated.
- ✓ The system must print out the result: success or fail.
- ✓ The program returns to the main screen.

5. **Report (100 LOCs)**

5.1. The ingredients are available. (25 LOCs)

- ✓ The system searches for the ingredients list, and returns all ingredients that have available status.
- ✓ Show the result list: code, name, and other attributes.
- ✓ The program returns to the main screen.

5.2. The drinks for which the store is out of ingredients. (50 LOCs)

- ✓ The system searches the menu and returns all drinks whose ingredients are unavailable.
- ✓ Show the result list: code, name, and other attributes.
- ✓ The program returns to the main screen.

5.3. Show all the dispensing drink. (25 LOCs)

- ✓ Show all the data in the dispensing drink collection on the screen.

- ✓ Sorted in ascending order by name
- ✓ The program returns to the main screen.

6. **Store data to files. (50 LOCs)**
    - ✓ Store the list of ingredients in the **Ingredients.dat** file.
    - ✓ The store list drinks information to the **Menu.dat** file.
    - ✓ Store list dispensing drink information to the **Order.dat** file.
    - ✓ Reload data when the program starts.

- The above specifications are only basic information; you must perform a requirements analysis step and build the application according to real requirements.
- The lecturer will explain the requirement only once on the first slot of the assignment.