LAB 211 Assignment

Type:	**Long Assignment**
Code:	**J1.L.P0001**
LOC:	**500**
Slot(s):	**N/A**

Title

PetStore Management

Background

Write a program to manage a pet store. The program implements the terminology of the Object-Oriented Programming (OOP) paradigm. OOP is one of the best choosing ways to design software programs.

In this assignment, we will use the Map<Key, Value> structure to store a collection of objects. The Key is the id of the object. The Value is the object. The Map structure is popularly used to manipulate data.

Program Specifications

Build the Pet Store project with the menu as follows:

1. Add a pet
2. Find a pet
3. Update a pet
4. Delete a pet
5. Add an order
6. List orders
7. Sort orders
8. Save data
9. Load data
10. Quit

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

Features:

Function 1. Add a pet - 50 LOC

- The application requires inputting a pet: id, description, import date, unit price, and category.
- The system will check the data validation with the following conditions:
  - All fields are not allowed null.
  - The id field must be unique.
  - The length of the description field must be from 3 to 50 characters.
  - The import date field must be a valid date.
  - The unit price field must be a positive number.
  - The category field must be one of the following: Cat, Dog, and Parrot.

- Add the pet should be stored to the collection of pets.
- The program should ask the user to choose add new pet continuously or going back to the main menu.

Function 2. Find a pet – 50 LOC

- The application requires inputting the pet id.
- If the pet does not exist, the message "The pet does not exist" is displayed. Otherwise, the system will display the pet.

Function 3. Update a pet – 50 LOC

- The application requires inputting the pet id.
- If the pet does not exist, the message "The pet does not exist" is displayed. Otherwise, the user can edit the pet.
- The system should show the result of this action with success or fail status.

Function 4. Delete a pet – 50 LOC

- The application requires inputting the pet's id.
- If the pet does not exist, the message "The pet does not exist" is displayed. Otherwise, the user can delete the pet.
  - The system must show the confirm message before delete action is taken.
  - The pet cannot be deleted if it is already in an order detail.
- The system should show the result of this action with success or fail status.

Function 5. Add an order – 50 LOC

- An order includes an order header and several order details. The order header includes the order id, the order date, and the customer name. The order detail includes the order detail's id, the pet's id, the quantity, and the pet's cost.
- The system will check the data validation with the following conditions:
  - All fields are not allowed null.
  - The id fields must be unique.
  - The order's date field must be a valid date.
  - The quantity field must be a positive integer.
  - The pet's cost field is the field calculated as follows: pet cost = quantity x unit price.
- The system will add the order to the collection of orders.
- The program should ask the user creating new order continuously or going back to the main menu.

Function 6. List orders – 50 LOC

- The application requires inputting a start and end date.
- List orders for that date range as below should be recommened.

LIST ORDERS FROM 10/01/2022 TO 10/31/2022

No.	Order Id	Order Date	Customer	Pet Count	Order Total
1	0006	10/01/2022	John Smith	3	$ 160
2	0007	10/15/2022	Bill Jamie	5	$ 220
3	0008	10/26/2022	John Smith	2	$ 60
	Total			10	$ 440

Function 7. Sort orders – 50 LOC

- The application requires inputting a sorted field (order id or order date or customer name or order total) and the sort order (ASC, DESC).
- The data grid is shown orders as below.

LIST OF ORDERS

Sorted by : Order Date
Sort order : ASC

No.	Order Id	Order Date	Customer	Pet Count	Order Total
1	0006	10/01/2022	John Smith	3	$ 160
2	0007	10/15/2022	Bill Jamie	5	$ 220
3	0008	10/26/2022	John Smith	2	$ 60
	Total			10	$ 440

Function 8. Save data – 50 LOC

- The system will store the collection of pets to the binary file with naming pets.dat.
- All the order information in the collection storage will be store the binary file orders.dat.

Function 9. Load data – 50 LOC

- The system can read pets from the binary file pets.dat to the collection structure.
- The system can also read pets from the binary file orders.dat to the collection structure.

Function 10. Quit – 50 LOC

- The application should allow the user quit the program.
  - The system must show the confirm message before quit action is taken.
  - The system must save data to files if data is changed.

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 in the first slot of the assignment.