LAB 211 Assignment

Type:	**Long Assignment**
Code:	**J1.L.P0004**
LOC:	**550**
Slot(s):	**N/A**

Title

The Zoo Management

Background

Mr. H plans to develop a program to manage animals in the zoo. Animals have many properties in common. In addition, it also has its own. This management program needs to have basic functions such as: add, edit, delete, search. The program must be designed so that adding a new animal is easy. Let's build your idea based OOP model.

Program Specifications

Build a management program. With the following basic functions

0. Build your data structure
1. Load data from file (Optional)
2. Add new animal
3. Update animal
4. Delete animal
5. Search animal
   - 5.1 Search by name
   - 5.2 Search by type
6. Show animal list
   - 6.1 Show by Type
   - 6.2 Show all
7. Store data to file (Optional)

Others- 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. All animal's information in the zoo is contained in file animals.txt.

In the current stage, the zoo has 4 groups of animals as follows:

- Group of 0-legged animals- snake has the properties such as no leg, black and white color, can creep, poisonous, 3 kg.
- Group of bipedal and flightless animals- penguin has the properties such as two legs, two swings, cute, color, can walk, make sound, 6kg.
- Group of bipedal and flying animals: pigeon has the properties such as two legs, two swings, heroic, can fly, 1kg.
- Group of 4-legged animals: tiger has the properties such as 4 legs, can run, can growl, danger, eat raw meat, arrive, 100kg.

Features:

This system contains the following functions:

Display a menu and ask users to select an option.

- **Function 0: Build the data structure**: **100LOC**
  - Classes, abstract classes, Interfaces.
  - Use only one collection to store animals.
- **Function 1: Load data from file-50LOC**
  - Load all data in the file into the collection.

- **Function 2: Add new animal- 50 LOC**
  - Create a submenu that allows the user to add animals to the zoo.
  - Remember that the constraints must be checked
  - Add the new animal to collection.
  - Ask to continuous create new animal or go back to the main menu.

- **Function 3: Update Animal- 50LOC**
  - Require enter the animal's id.
  - If animal does not exist, the notification "Animal does not exist". Otherwise, user can start input new information of animal and update.
  - If new information is blank, then not change old information.
  - Then system must print out the result of the updating.
  - After updating, the program returns to the main screen.

- **Function 4: Delete Animal- 50LOC**
  - User can delete any animal in the zoo.
  - Before the delete system must show confirm message.
  - Show the result of the delete: success or fail.
  - After delete, the program returns to the main screen

- **Function 5: Search animal**
  - Create a submenu that allows the user to select way to search: search by name or by type.

- **F.5.1: Search by Name– 50 LOC**
  - User input the text want to search.
  - The system will search in the zoo, and return all animals that has name contain the search string.
  - Show result list: all information of animals.

- **F.5.1: Search by Weight– 50 LOC**
  - The user enters the upper and lower bound of the desired weight.
  - The system searches the zoo, and returns all animals whose weights meet the search criteria.
  - Show result list: all information of animals.

- **Function 6: Show animal list**
  - Create a submenu that allows the user to select way to show: show all or by type.

- **F.6.1: Show by type– 50 LOC**
  - User select animal's type want to show.
  - The system will show this list of animal belong that type.
  - Show result list: all information of animals.

- **F.6.1: Show all– 50 LOC**
    - The system will show this list of animal in the zoo.
    - Show result list: all information of animals.

- **Function 7: Store data to file-50LOC**
    - Store data in collection to file.

- 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.