**5University of Newcastle**

**School of Electrical Engineering and Computing**
**SENG1110/6110 Programming Assignment 1 – Semester 1, 2018**
**Due:** By electronic submission (Blackboard) by 11:59pm on **Fri 01-Jun.**

# Vet clinic management – part 2

## Introduction

The objective of this assignment is to extend the implementation of Assignment 1 using **arrays** and **external files**.

## Before you start

Carefully read the problem description below. Make sure that you have all the information necessary to start the program. Do not assume what is necessary. There is a discussion board forum: assignment 2. Post your questions there and check it regularly.

## Problem Description

Your task in this assignment is to extend your vet clinic management program from Assignment 1, by **adding** the following functionality:

1. Will allow the user to **input any number of pets and doctors**.
2. Will allow the user to **edit the information of a pet**. The input will be the name of the pet. If the name does not exist, the program should show a message. The user will be able to edit any information of the pet, except its name.
3. Will allow the user to **read all the information of pets and doctors from a file** called "VetManagement.txt"
4. Will allow the user to **save all the information of pets and doctors to a file** called "VetManagement.txt".

Below, the functionality from assignment1 that should continue in the assignment2.

5. Will allow the user to **enter a new doctor** into the vet clinic. Each doctor will have the following information:
    - `name` – the name of the doctor.
    - `specialisation` – the specialisation of the doctor (it must be "dog" or "cat", if not, the program should show a message and ask again)
6. Will allow the user to **enter a new pet** into the vet clinic. Each pet will have the following information:
    - `type` – the type of the pet. It can be only "cat" or "dog". if not, the program should show a message and ask again.
    - `size` – the size of the pet. It can be only "small", "medium" or "large". if not, the program should show a message and ask again
    - `name` – the name of the pet.
    - `weight` – the weight of the pet. It should be a positive number. if not, the program should show a message and ask again.
    - `age` – the age of the pet. It should be a positive number. if not, the program should show a message and ask again.
    - `doctor` – the doctor of the pet. The doctor of a new pet should hold "no doctor assigned".
7. Will allow the user to **delete a pet** from the vet clinic. The input will be the name of the pet. If the name does not exist, the program should show a message.
8. Will allow the user to **delete a doctor** from the vet clinic. The input will be the name of the doctor. If the name does not exist, the program should show a message.
    - Before deleting the doctor, all pets that are treated with this doctor should have doctor updated to "no doctor assigned"
9. Will allow the user to request the **list of doctors** in the vet clinic and all information of each doctor. If there are no doctors, the program should show the message "no doctors".

10. Will allow the user to request the **list of pets** in the vet clinic and all the information of each pet. If there are no pets, the program should show the message "no pets".
11. Will allow the user to request the **list of pets** under a specific doctor (the user will input the name of the doctor). If the name does not exist, the program should show a message.
12. Will allow the user to **assign a doctor to a pet**. The user will input the name of the pet and the name of the doctor. If the pet already has a doctor assigned, the program will ask the user if (s)he would like to change the doctor. If the doctor does not have the right specialisation, the program will show a message. If the pet or the doctor do not exist, the program will show a message.
13. Will allow the user to **analyse a pet**. The user will input the name of the pet and the program will give all the details and will tell if the dog or cat is overweight. Suppose a cat is considered overweight if:
    - Small and weight greater than 4kg
    - Medium and weight greater than 6kg
    - Large and weight greater than 8kg

    Suppose a dog is considered overweight if
    - Small and weight greater than 6kg
    - Medium and weight greater than 9kg
    - Large and weight greater than 12kg

    If the pet does not exist, the program will show a message.

Your program must give appropriate messages to the user on an attempt to:

- add a pet or doctor that already exist (note that 2 pets/doctors are identical if the names are the same).
- Note that Strings (name of the doctor and pet) must not be case sensitivity. For example, "Benny", "benny", "BENNY" or even "BennY" are the same pet.

**Program Requirements:**

Your program should consist of three classes, which stores the following data:

- `Doctor.java` – stores the following details about a doctor.
  - `name` – String - the name of the doctor.
  - `specialisation` – String - the specialisation of the doctor (it can be "dog" or "cat")
- `Pet.java` – store the following details about a pet.
  - `type` – String - the type of the pet. It can be only "cat" or "dog".
  - `size` – String - the size of the pet. It can be only "small", "medium" or "large".
  - `name` – String - the name of the pet.
  - `weight` – double - the weight of the pet.
  - `age` – int - the age of the pet.
  - `doctor` – String - the doctor of the pet.
- `Clinic.java` – provides the information of all doctors and pets in the clinic.
  - `doctorList[]` – an array of Doctor objects. The initial size should be 3.
  - `petList[]` – and array of Pet objects. The initial size should be 3.

You should continue to apply the principles of encapsulation to your classes, and also write **two** constructors in the classes Pet and Doctor. The first constructor should be the default one, i.e., no parameters. The second constructor will have as parameters values to the instance variables. In the `Clinic` class, you should have a method for resizing the `doctorList` and `petList` arrays.

All classes need to have methods that provide the functionality outlined in the problem description. The only class which should have a **main method** is `Clinic.java`. The class `Clinic` also will be the only one that will receive inputs and show outputs. You can use TIO or GUI (it is your choice).

The format for the file VetManagement.txt is provided in Blackboard. Your program should be able to read this file, and output to file in the same format. Note that it is just a text file.

Your solution must be your own work. **Marks** will be awarded for: layout, both visual (variable names, indentation) and structural (scope of variables, use of methods); documentation (comments); and ability of the submission to perform as specified. A more detailed marking schema will be available in Blackboard.

**What to submit.**

You should submit the Java program (`Doctor.java`, `Pet.java` and `Clinic.java`) and the assignment cover sheet (available in Blackboard) in a compressed .zip folder, electronically under the **Assignment 2** link in Blackboard. No `.class` files should be included in your submission, only `.java` files.

**Extra Work for SENG6110 students**

The functionalities 9 and 10 of your program should give the option to the user to choose the output to be in alphabetical order by name.

**SENG1110 students that implement the SENG6110 task will receive extra 15 marks (the total mark will not be more than 100).**

**Late Penalty and adverse circumstances**

Note that your mark will be reduced by 10% for each day (or part day) that the assignment is late. This applies equally to week and weekend days. You are entitled to apply for special consideration because adverse circumstances have had an impact on your performance in an assessment item. This includes applying for an extension of time to complete an assessment item. See https://www.newcastle.edu.au/current-students/learning/assessments-and-exams/adverse-circumstances for more details.

**On Blackboard you will find a new forum in the discussion board: "assignment2". Any question about the assignment 2 can be posted there. Check this forum regularly.**

Prof Regina Berretta - 2018