

PROGRAMMING ASSIGNMENT 4:

In this assignment you will be doing a simulation of multiple humans and dogs. You can choose to change the types of classes (so make it a cat, horse, snake or something if you like), but ensure that you are still doing an equal amount of work.

In this assignment, you will have 4 classes.

- 1) A class Mammal, which will have attributes of name, weight, hunger, and sex.
- 2) A class Human, which extends Mammal, having attributes of money, dogfood, and has-a certain number of dogs specified on creation (so ArrayList of Dog). Your methods for the human class should be walks, feeds, bathes, toString, passTheTime and buysFood. These methods should use getter and setter functions to set the values as mentioned below.
- 3) A class Dog, which extends Mammal, having attributes of hunger(1-100), fun(1-100), cleanliness(1-100) and loyalty(1-100) and a method toString. Loyalty should be a private method that can only be changed by a dog and is always kept at the maximum value of the two most important attributes for a dog ($\max(\text{fun}, 100 - \text{hunger})$). Make sure to use getter/setter methods to enforce that loyalty always is set to this.
- 4) A class that uses these classes to run our simulation

In your main method, you should start by prompting how many humans should exist. Create these humans and prompt for names using an ArrayList or array. Each human should have from 1-2 dogs, you can either prompt for this or use random numbers. You must then prompt for each dog's name. Other attributes you can generate randomly.

The methods that allow you to interact with the dog are largely up to you, but some ideas would be something like the following:

Any dog walked should have its fun increased by 50, cleanliness decreased by 10%, hunger increased by 10%.

Any dog bathed, should have its fun decreased by 10% and cleanliness increased to 100.

Any dog fed should have its hunger increased by 50, cleanliness decreased by 10%, fun increased by 10%. This should also result in -1 food for each dog.

The method buyFood should result in 50 more food added, with 10 less money.

The method passTheTime should result in all dogs for that human having -1 less fun (they got bored) and an increase in 10 money (the humans made some money).

Once your humans and dogs are created, enter a loop. Inside the loop:

- 1) Each iteration of the loop, call the toString methods of all the Humans. This toString method should display the attributes of the humans and also call the toString methods of their dogs.
- 2) Then prompt for each human in turn to perform an action. This action could be walk, feed, bath his dogs or buysFood. Perform that action. Also allow the user to exist the program at this time.
- 3) Before starting a new iteration of the loop, call passTheTime with all human objects.
- 4) If at any time the loyalty of a dog hits 0, the program should terminate because a dog ran away or died (possibly notify PETA).

Feel free to mess around with the internal values of the different methods if you want, but make sure you at least do the minimum that I requested.

For submission, please archive together your 4 java files and submit them together. I will want UML class diagrams for your classes, both showing the prototypes of the methods and their hierarchy. This can be a separate file if you want.