DogsRUs Project

This project is a mini-assignment which was part of my object-oriented (Java) module in the second semester of my first year at university. I received 93% for my submission. The assignment was worth 16% of my overall grade for the module, and we were tasked with improving a program that would be used for organising the shelter and keeping track of all the different animals housed at Dogs RUs Kennels.

The main tasks of the assignment were to add inheritance and generally develop the code further by fleshing out the existing classes. The inheritance was to be added through the addition of an Animal class that would be the superclass of both an existing Dog and a new Cat class.

To prevent code duplication across my application and to ensure efficiency I created an AnimalType enum and a TakeInput class, both of these classes allowed me to eliminate code duplication. The AnimalType enum has two possible values: DOG or CAT and is used as a parameter to methods and the outcome of the method changes depending on the value of that object. This could have possibly be done by detecting the type of object being considered but I was not able to execute that. The TakeInput class, on the other hand, gets input from the user and validates that input. Putting these methods in a separate class from the main class meant that the class could easily be reused in the future and trimmed down the size of DogsRUsApplication.

I made sure to rework all the existing code to either make it more efficient or more consistent. Including the pre-existing comments.

Throughout this project, I looked to create the most efficient, consistent, and robust application that I could, with the best feature set. And I believe the final application was of a high quality, but could still be improved.

# Class Details

## Animal

Superclass to both Cat and Dog holds all generic instance variables and methods for any other Animal classes - currently Animal is a superclass to both Cat and Dog. Included in those instance variables is an ArrayList originalOwners of type Owner - this variable is used to store all the original owners for this Animal and also connects the two classes.

One special feature of this class is that implements the Comparable interface and includes a compareTo method, which enables a comparison between the names of two animals.

## AnimalHousing

This was originally the Kennel class, but I updated the name of the class and the features of the class to be more inclusive since cats have also been introduced to the shelter, and I do not expect cats and dogs to be sharing the same housing. I use this class in DogRUsApplication to create a kennel and a cattery object.

The class has the highest level save and load methods and is thus used to save all the information relating to each AnimalHousing, which includes information on the animals and owners in each housing, in a text file.

## AnimalType (enum)

This is an enum class that is used to create objects that can only contain 'CAT' or 'DOG'. These objects are used as parameters in several methods to reduce code duplication, but still, permit different outcomes depending on the animal's type.

## Cat and Dog

Both of these classes are an extension of the Animal class, but they both have some specific additions in instances variables and methods. Cat has a boolean canShareRun instance variable unique to it since cats are more likely to fight with other cats. This extra instance variable unique to Cat means that it needs its own getter and setter, and some of the existing Animal methods need to be refined - they will still retain all of their original functionality, but will also include one to a few additional statements. This is the same for Dog, which has two unique instance variables that change the data it holds and the functions of its methods.

## DogsRUsApplication

This class includes the main and runs the whole application. This class provides the user with a menu and depending on the selection made by the user - the appropriate operations will be executed. A large proportion of the utility methods in this class use AnimalType objects as parameters. Meaning one method can be used for either type of animal, but will still provide an outcome unique to that animal type.

## Owner

This method holds all of the relevant instance variables and methods associated with an owner.

## TakeInput

This class is used to scan in and validate input from the user - making sure it is of the right type and the input provided is not an empty piece of input. This class is used within DogsRUsApplication and AnimalHousing to handle all user input.