Programming Assignment #3: Java

Due at the beginning of the class on December 5

This assignment requires you to develop an object oriented software system in Java that will keep track of pets treated and boarded in an animal hospital. Detail class specifications (data members, methods and access modifiers) are described below. Please note the following requirements:

- You need to implement each class in a separate file.
- While implementing the design you may want to follow the order the classes are specified below and test each class individually.
- In the description below, items that are *italicized* must appear exactly as they are stated. Our test program will fail to run your code if you change them, even by a character or a case. This may result in severe penalty, so please be very careful with those items.

Class: Pet (File name: Pet.java)

The class should have the following three private data members, Pet name (a String), owner name (a String), and color (a String), and one protected data members for *sex* (an integer, but it will only hold one of the following four public static final int values: *MALE*, *FEMALE*, *SPAYED* and *NEUTERED*. You should define these four static finals in your class).

Following are the **public** methods that this class should provide:

Pet (String name, String ownerName, String color); *//Constructor*
String getPetName();
String getOwnerName();
String getColor();
void setSex(int sexid);
String getSex(); // Should return the string equivalent of the gender, e.g the string “male” etc.
String toString();
// Should return the name, owner’s name, age, color, and gender (use getSex());

A Sample (preferred) return value by toString is as follows:

Spot owned by Mary
Color: Black and White
Sex: Male

Interface: Boardable (File name: Boardable.java)

This interface, should include the following **public** methods:

void setBoardStart(int month, int day, int year);
void setBoardEnd(int month, int day, int year);
boolean boarding(int month, int day, int year);

See the Cat and Dog classes for what these methods should do when implemented. Note, the month will be in the range 1-12, day in the range 1-31, and year will be a four digit number.

Class: Cat (File name: Cat.java)

This class should extend the Pet class and implement the Boardable interface. In addition to the data members and methods inherited from Pet, the Cat class should have a private *hairLength* data member, which is a string.

Following are the **public** methods that this class should provide

Cat (String name, String ownerName, String color, String hairLength);
// Do not forget to call super.
String getHairLength(); // returns the string hairLength
String toString()
/* method that returns a String that identifies the pet as Cat and returns a complete description of the cat, **including** the values stored in the Pet parent class.*/

A Sample (preferred) return value by toString is as follows:
CAT:
Tom owned by Bob
Color: black
Sex: spayed
Hair: short

In order to implement the *Boardable* interface define new data members to store the boarding start and end dates, implement the *setBoardStart* and *setBoardEnd* methods to store values for these data members. Also implement the boarding method to return true if the given data is between the start and end dates, otherwise it returns false. Note: You should also return true if the given date is equal to the start or end date.

Class: Dog (File name: Dog.java)

This class should extend the Pet class and implement the Boardable interface. In addition to the data members and methods inherited from Pet, the Dog class should have a private *size* data member, which is a string.

Following are the **public** methods that this class should provide:

Dog (String name, String ownerName, String color, String size);
// Constructor must set the size. Do not forget to call super.
String getSize(); // returns the string size
String toString();
/* method that returns a String that identifies the pet as Dog and returns a complete description of the dog, **including** the values stored in the Pet parent class. */

A Sample (preferred) return value by toString is as follows:
DOG:
Spot owned by Susan
Color: white
Sex: spayed
Size: medium

In order to implement the *Boardable* interface define new data members to store the boarding start and end dates, implement the *setBoardStart* and *setBoardEnd* methods to store values for these data members. Also implement the boarding method to return true if the given data is between the start and end dates, otherwise it returns false. Note: You should also return true if the given date is equal to the start or end date.

Class: Bird (File name: Bird.java)

This class should extend the Pet class. In addition to the data members and methods inherited from Pet , the Bird class should have a private *boolean* data member called *feathersClipped*.

Following are the **public** methods that this class should provide:

Bird(String name, String ownerName, String color) ;
    /* Constructor should initialize the feathersClipped data member to false. Do not forget to call super. */
boolean clipped(); // returns the value of feathersClipped
void setClipped(); // sets the the value of featherClipped to true
String toString()
    /* method that returns a String that identifies the pet as Bird and returns a complete description of the bird, **including** the values stored in the Pet parent class. */

A Sample (preferred) return value by toString is as follows:
BIRD:
Poly owned by Steve
Color: green
Sex: spayed
Feather clipped: yes

Class: AnimalHospital (File name: AnimalHospital.java)

This class does not extend any other classes or implement any interfaces, although it will use the other classes you have developed. You should provide the following **public** methods:

AnimalHospital(String inputFile);
void printPetInfoByName(String name);
void printPetInfoByOwner(String name);
void printPetsBoarding(int month, int day, year);

The constructor should take a file name as an argument and read in the file information. The input file will consist of a series of records for different animals, where each record consists of multiple lines. The first line will be either the string “CAT”, “DOG” or “BIRD” to signify the kind of pet record that follows. The information following that “flag” element will contain the pet’s name, owner’s name, its color, its gender, its hair length if it is a cat and its size if it is a dog. Birds will not have any additional attribute, since the clipped information is set as default. Note the gender line will consist of one of the following four strings: “male”, “female”, “spayed”, “neutered”. The last line of the file will be the string “END”. You should read each line of the file, create objects for each pet, and add them to an appropriate data structure.

We have provided a sample input file pets.data that can be downloaded from http://www.cse.lehigh.edu/~abq2/pets.data. We have also provided a LineReader class that makes it easy to read the text file one line at a time. The code is available at http://www.cse.lehigh.edu/~abq2/LineReader.java. In order to use this class, you must call its constructor with the name of a file to open, and then use the nextLine() method each time you want a line from the file. For example:

```
LineReader myfile = new LineReader("pets.data");
String firstLine = myfile.nextLine();
   /* reads the first line from "pets.data". Calling it again will read the next line, etc. */
```

The various print methods should do the following:

- printPetInfoByName() will search the list of pets for every pet of a given name, and print the pet's information, using the toString() method.
- printPetInfoByOwner() will search the list of pets for pets owned by the given person and print the pet's information for every match, using the toString() method.
- printPetsBoarding() will search the list of pets for every pet boarding at the given time and print the pet's information for every match, using the toString() method.Note that to test this method in a reasonable manner you will need to invoke *setBoardStart* and *setBoardEnd* a few times on some of the cats and dogs that you read in, since that information is not in the file and is not generated by the constructors.

Evaluation Note (important)

To evaluate your work, we will create a java application and in that application we will create an object of type AnimalHospital by passing it a file name. Then we will test your public methods of various classes. You may want to create a test program in the same manner. You may assume that all inputs will be in their valid ranges.

Submission instructions:

You will need to submit both electronic versions of your program file and hardcopy of your *source* code

- Zip all your java files (Pet.java, Boardable.java, Cat.java, Dog.java, Bird.java, and AnimalHospital.java). You may name it pa3.zip.
- This zip file must be submitted by using the course webpage on the Blackboard Learning System (see https://ci.lehigh.edu/). From the CSE 262 page, select Assignments, and then click on "View/Complete". You will then be able to attach each of your file for submissions.