Programming Assignment 3

Due Tuesday, June 28, 2016 at 11:59 P.M.

In this assignment you are tasked with creating a Java package. The package will be used by an external program that you do not have access to. Therefore you must follow all specifications exactly. A missing capitalization or incorrect method signature will cause your package to not work with the external program. Please follow all instructions exactly. If your package fails to work with the external program you will lose 15 points and as your code is evaluated you will lose additional points for each specification not followed.

Students please realize this is not a small assignment, you will need to start early. Do not try to code everything at once. Read the whole assignment, then break it down to smaller pieces. And then code each piece, make sure to check your code often. You do not want to finish coding everything and then have several errors, without knowing where to start. And do not expect the TA to debug the code for you. When you email for help be very specific. Don't code everything, then find one error at the end, and expect the TA to inspect all of your code. You need to show effort that you are trying to solve the error before asking for help.

Description

- The final goal is to have a simple pet game.
- You are tasked with creating a package that will be used by an executable.
- The executable for the game will be created by the TA.
- This executable will be delivered to you on Monday, June 27.
- The code will involve use of arrays, interfaces, inheritance, composition and file i/o.
- Please review these concepts before continuing.
- You must design a class that can read a text file. Specifications below.
- You must design three pet class types: `Duck`, `Hamster` and `Pig`. Specifications below.
- The executable will do the following
    - Use your file reader class to get pet details from text file.
    - It will create pets based on these values.
    - Then the game will start.
- The game itself is very simple
    - It will randomly make your pets hungry, thirsty and/or bored.
    - When they get hungry/thirsty/bored you must feed/give water/play with them.

All your java files must be in a package "`petfarm`" (take note that the name is all lowercase). Eclipse users make sure you add a new package, `petfarm`, before creating any java files.

Before making the `Pet` base class, you will need an interface and a new class. Following are the specifications for those two.

Interface Teachable

This interface will be used by the derived classes of the Pet class.

- Access level of Teachable is public.
- It is a simple interface that has one method called teach
- Access level of teach is public. Return type is void
- It accepts one parameter of type String[], name it phrases.

Class Attribute

This class will be used as a part of the Pet class. Note that there is a private method. This is intentional so make sure that your code reflects this.

- Class access level is public.
- Two fields
    - private int rate: this int will range from 1 and 3. Higher values signal more frequent need.
    - private int status: this int must be between 0 and 5. The modifying methods specified below, will handle this.
- Single Constructor
    - Access level is public
    - Takes one parameter of type int that initializes the rate value. Use the setRate method specified below.
    - Initialize status to 5
- private setRate method
    - return type void
    - one parameter of type int
    - sets rate equal to the parameter passed
- public getRate method
    - return type int
    - no parameters
    - returns current value of rate
- public getStatus method
    - return type int
    - no parameters
    - returns current value of status
- public incStatus method
    - return type boolean
    - if value smaller than 5, add 1 and return true
    - if value already 5, then do not change and return false
    - goal of this method is to ensure status is modified correctly. If it isn't then it will return false, indicating something went wrong.
- public decStatus method
    - return type boolean
    - if value larger than 0, subtract 1 and return true

  - if value already 0, then do not change and return false
  - this method also ensures proper modifications to status like incStatus.

Class Pet

This class will make use of the Attribute class that you created. Please pay special attention to the access levels of all member fields and methods. THEY ARE IMPORTANT.

- Pet is an abstract class and the access level is public
- Fields
  - private Attribute hunger
  - private Attribute thirst
  - private Attribute bored
  - private String[] phrases
- Constructor
  - Access level protected
  - It is protected to ensure that only derived classes can use this constructor.
  - 3 parameters: int hungerRate, int thirstRate, int boredRate
  - Initialize the Attribute fields with their respective parameter
  - do not initialize phrases field.
- public methods
  - getHunger: returns the hunger object
  - getThirst: returns the thirst object
  - getBored: returns the bored object
  - getPhrases: returns the phrases object
- protected method
  - setPhrases: one parameter of type String[], sets phrases to that parameter
  - this is protected to ensure only derived classes can use this method.

The derived classes of Pet

All the derived classes are identical. If there is a difference they will be detailed in the specification below.

- Each derived class have access level public.
- Each derived class implements Teachable.
- The class names are: Duck, Hamster, Pig.
- The classes have one additional field called name, and it is of type String
- Each class has a single constructor
  - public access level
  - takes 4 parameters
    - String name
    - int hungerRate
    - int thirstRate
    - int boredRate
  - use Pet's protected constructor to initialize the 3 attributes

    - then initialize `name` to the passed `String` parameter.
- The `public teach` method
    - simply pass the parameter of `teach` method into `setPhrases` method of `Pet` class.
- Also include a `getName` method
    - access level is `public`
    - return the `name` object

Class `Petfiles`

This class is what will read a provide text file. The text file will include information for all pets of the pet farm. Following is only an EXAMPLE of the text file. This is not the actual text file that will be used to grade your assignment. Just an example:

```
2
Duck
Rover
2
2
3
Pig
Wilbur
3
1
1
```

The first line of the file will always be how many `Pet` types are in the file.
Then 5 lines will be used to provide information on each `Pet`.

- Line 1: will be type of `Pet`
- Line 2: is name of the `Pet`
- Line 3: is the `hunger` rate of the `Pet`
- Line 4: is the `thirst` rate of the `Pet`
- Line 5: is the `bored` rate of the `Pet`

Use these values to initialize the `Pet` array with the individual `Pet`s given.

Specifications of class

- `public` access level
- One field: `private Pet[] pets`
- single `public` constructor
    - It takes one parameter of type `String`.
    - This is the `String` of the filename.
    - Use it to create an object to read from that file.
    - Check to make sure that the file is properly opened. Otherwise print a message that the file opening failed.
    - Initialize `pets` to the correct size

  - The use a `for` loop to:
    - read from file 5 lines
    - create object based on those 5 lines
    - set the corresponding position in `pets` to that new `Duck`, `Hamster` or `Pig`.
- single `public` method
  - name is `getPets`
  - return type is `Pet[]`
  - simply return the `pets` object

If any of the instructions are unclear, do not hesitate to email the TAs. There will be no tolerance for misunderstanding the instructions. So please reach out to the TAs so that all the instructions are clear to you.

Special instructions for the submission will be included with the rubric for grading. These will be posted to blackboard in a separate file.

Good luck and have fun!