COMP201- Object Oriented Programming

Fall 2016

HOMEWORK3

In this homework, you are asked to write an application for a weird zoo.

**Required Classes:**

The class diagram for the application is given in Canvas under the file/week3 director. You can also see a very small picture of it below:



According to the class diagram;

* Cats, dogs, fishes and birds are pets.
* Eagles and snakes are wild animals.
* Both pets and wild animals are animals.
* All the instance variables are defined as private. Also, there are set/get methods for all those variables.
* There is **only** one constructor in the Animal class, which takes a string parameter to set it as the name of the animal. Also, this constructor sets the numberOfLegs to 4 and numberOfWings to 0 as default values.
* In the constructors of subclasses, numberOfLegs is set to 2 for birds and eagles and it is set to 0 for fishes. For numberOfWings, birds and eagles set it to 2 in the constructor.
* For wild animals, there is a habitat variable defined and it is set to “forest” by default. For eagles, it is set to “mountain”.
* For pets, there is a costOfFeed variable, which is set to 0 at the initialization time. This variable defines the total money that the pet-owner spent for each pet.
* The definition of methods are below:
  + talk()
    - For animals this method prints “<Silence>”unless it is defined otherwise.
    - For Cat, Dog and Bird classes, this method prints the sound that the animal makes (i.e. MEOW, BARK, CHIRP)
  + fly()
    - This method prints the name of the animal and “cannot fly” by default unless it is defined otherwise.
    - For birds and eagles, this method prints the name of the animal and “is flying”.
  + run()
    - This method prints the name of the animal and “is running” by default unless it is defined otherwise.
    - For birds, eagles, fishes and snakes, this method prints the name of the animal and “cannot run”.
  + feed()
    - This method takes an integer value as a parameter and adds it to the costOfFeeding variable.
  + cleanHimself()
    - Cats can clean theirselves. This method prints the name of the cat and “ is clean now” to the screen.
  + walk()
    - Dogs need to get walk everyday. This method prints the name of the dog and “walked” to the screen.
  + swim()
    - Fishes can swim and this method prints the name of the fish and “is swimming” to the screen.
  + crawl()
    - Snakes crawl and this method prints the name of the snake and “ is crawling” to the screen.

**Program Execution:**

The program consists of two levels:

**Level-1:** In the first level, the user enters the names of the animals and their species. To finish the first level, the user enters “END”.

**Level-2:** After setting all the animals in the first level, the user enters commands in the second level. The program runs until the user enters “EXIT”.

The list of the commands are below :

* talk <nameOfTheAnimal> : the program calls the talk method for the animal.
* fly <nameOfTheAnimal> : the program calls the fly method for the animal.
* run <nameOfTheAnimal> : the program calls the run method for the animal.
* legCount <nameOfTheAnimal> : the program prints the number of legs of the animal.
* wingCount <nameOfTheAnimal> : the program prints the number of wings of the animal.
* feed <nameOfThePet> <Amount>: the program calls the feed method for the Pet for the given amount. Note that this command can be used only for Pets.
* costOfFeed <nameOfThePet>: It prints the money spent for feeding the animal. The command is used only for pets.
* clean <nameOfTheCat> : the program calls the cleanHimself method for the animal. Note that this method can only be used for cats.
* walk <nameOfTheDog> : the program calls the walk method for the animal. Note that this method can only be used for dogs.
* swim <nameOfTheFish> : the program calls the swim method for the animal. Note that this method can only be used for fishes.
* crawl <nameOfTheSnake> : the program calls the crawl method for the animal. Note that this method can only be used for snakes
* habitat <nameOfTheWildAnimal> : the program prints the habitat of the animal to the screen. The command is only used for WildAnimals.
* exit: The program terminates.

An example of the program execution is given below:

Please enter <Name> <Type> of animals. Enter END to pass to second level.

kiwi Bird

charlie Eagle

mango Fish

pepper Dog

angle Cat

tweety Bird

bob Dog

ziggy Snake

END

Enter <command> <animal> [<option>]. Enter EXIT to exit the program.

talk angle

MEOW

fly mango

mango cannot fly

run pepper

pepper is running

legCount ziggy

0

wingCount kiwi

2

feed tweety 100

feed tweety 50

costOfFeed tweety

150

clean angle

angle is clean now

walk bob

bob walked

swim mango

mango is swimming

crawl ziggy

ziggy is crawling

habitat charlie

mountain

EXIT

**Programming Tips:**

* In this homework, you need to create an array of Animal. You can set the size of array to 100. You can use the following code-line for this purpose.

Animal[] animals = new Animla[100];

* For instance, to create a cat in this animal array, please remember that the following code is valid.

Animal animal;

animal = new Cat(“angle”);

* When you need to call a method belongs to only to the subclass, you need to do a typecasting.

**Restrictions:**

* Animal class has only one constructor and takes a string parameter.
* All the definitions given in the class diagram should be applied as it is. For instance, there is no swim method in the Animal class.
* You are allowed to have only one array in the main method and the type of the array should be Animal.
* The inputs to the program will be error free.

**Submission Rules:**

* Make a directory and name it as name\_surname\_number.
* Put your .java files (Main.java, Animal.java, Pet.java, Bird.java, Cat.java, Dog.java, Fish.java, WildAnimal.java, Eagle.java, Snake.java) under the directory.
* Compress the directors (use zip or rar)
* Upload your homework via canvas.