

Programming I - Assignment 8

Student Name: Evelyn Toledo Lally

Student Number: 22102020

Class Animal

```
import java.io.Serializable;
import java.lang.Comparable;

public class Animal implements Comparable<Animal> , Serializable {

    //Data members
    private String name;
    private String breed;
    private int size;

    //Default Constructor
    public Animal() {
        this("Unassigned", "Unassigned", 0 );
    }

    // Constructs a new Animal object with passed name, breed, and size.
    public Animal(String name, String breed, int size) {
        this.name = name;
        this.breed = breed;
        this.size = size;
    }

    //Override the compareTo() using name instance variable for comaparison.
    public int compareTo(Animal a) {
        return this.name.compareTo(a.getName());
    }

    //getters
    public String getName() {
        return this.name;
    }

    public String getBreed() {
        return this.breed;
    }

    public int getSize() {
        return this.size;
    }

    //Setters
```

```

public void setName (String name) {
    this.name = name;
}

public void setBreed (String breed) {
    this.breed = breed;
}

public void setSize (int size) {
    this.size = size;
}

//Override toString()
public String toString() {
    return "\nName: " + getName() + "\n"
    + "Breed: " + getBreed() + "\n"
    + "Size: " + getSize() + "\n";
}

} // end of class

```

Subclass Cat:

```

public class Cat extends Animal {

    //Constructor extends from super class Animal
    public Cat (String name, String breed, int size) {
        super(name, breed, size);
    }

    //Sound associated with cats
    public String sound() {
        return "Meow";
    }

}

} //end of class

```

Subclass Dog:

```
public class Dog extends Animal {

    //Constructor extends from super class Animal
    public Dog(String name, String breed, int size) {
        super(name, breed, size);
    }

    //Sound associated with dogs
    public String sound() {
        return "Woof Woof";
    }

} //end of class
```

Animal Driver Class:

```
import java.util.*;
import java.io.*;

public class DriverAnimal {

    public static void main (String[] args) {

        //Create and populate a linked list with Dog and Cat objects
        List<Animal> animals = new LinkedList<>();
        animals.add(new Cat("Bob", "Siamese", 3));
        animals.add(new Cat("Chichi", "Persian", 4));
        animals.add(new Dog("Lola", "Poodle", 2));
        animals.add(new Dog("Julio", "Bulldog", 5));

        //Display the contents of the list
        System.out.println("\nUnsorted Animal List:");
        System.out.println(animals.toString());

        //Sort the list by name
        System.out.println("\nSorting list by Name:");
        Collections.sort(animals);
        System.out.println(animals.toString());

        // Inner class implements Comparator by Breed
        class AnimalCompare implements Comparator<Animal>{
            public int compare(Animal one, Animal two){
                return one.getBreed().compareTo(two.getBreed());
            }
        }
    }
}
```

```

// AnimalCompare instance
AnimalCompare animalCompare = new AnimalCompare();
System.out.println("\nSorting by Breed:");
Collections.sort(animals, animalCompare);
System.out.println(animals.toString());

// Inner class implements Comparator by size
class AnimalCompareSize implements Comparator<Animal>{
    public int compare(Animal one, Animal two){
        return
Integer.valueOf(one.getSize()).compareTo(Integer.valueOf(two.getSize()));
    }

}

// AnimalCompareSize instance
AnimalCompareSize animalCompareSize = new AnimalCompareSize();
System.out.println("\nSorting by Size:");
Collections.sort(animals, animalCompareSize);
System.out.println(animals.toString());

//Serialize the list, then deserialize and display the list.
serialise(animals);
deserialise("AnimalList.dat");

} //end of main()

//Serialise method()
public static void serialise(List<Animal> list) {
    try {
        // create a connection stream (write bytes)
        FileOutputStream fileStream = new
FileOutputStream("AnimalList.dat");

        // create a chain stream (allows objects to be written to a stream)
        ObjectOutputStream os = new ObjectOutputStream(fileStream);
        // call writeObject() on the Object stream
        os.writeObject(list);

        os.close();

    }catch (Exception e) {
        e.printStackTrace();
    }
}

//Deserialise method()
public static void deserialise(String fileName) {
    try{
        FileInputStream fileStream = new FileInputStream(fileName);

        ObjectInputStream os = new ObjectInputStream(fileStream);

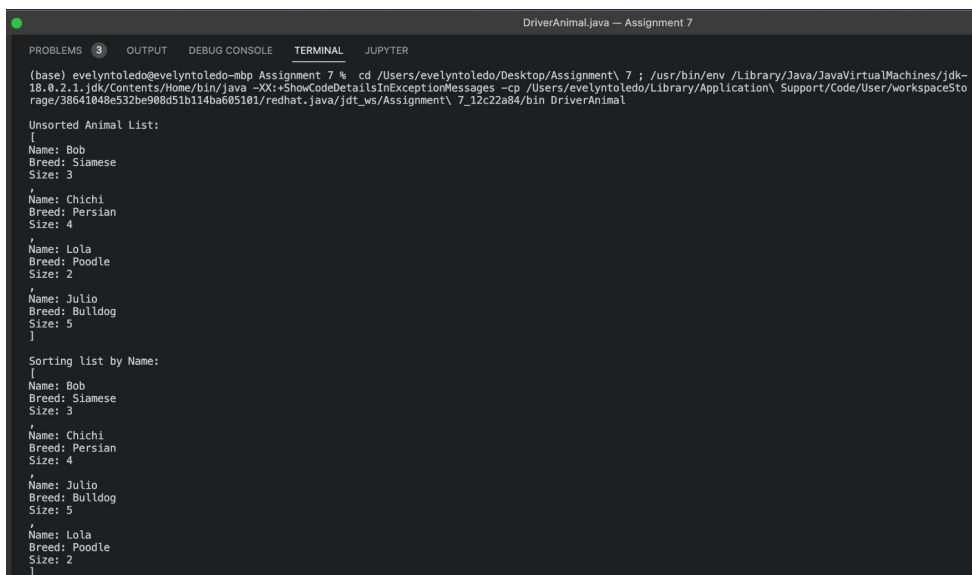
```

```

        List<Animal> animals = (List<Animal>)os.readObject();
        System.out.println("\nDeserialised:");
        for (Animal element:animals){
            System.out.println(element.toString());
        }
        os.close();
    } catch (Exception e) {
        e.printStackTrace();
    }
}
} //end of class

```

Output:



```

DriverAnimal.java — Assignment 7
PROBLEMS 3 OUTPUT DEBUG CONSOLE TERMINAL JUPYTER
(base) evelyntoledo@evelyntoledo-mbp Assignment 7 % cd /Users/evelyntoledo/Desktop/Assignment\ 7 ; /usr/bin/env /Library/Java/JavaVirtualMachines/jdk-18.0.2.1.jdk/Contents/Home/bin/java -XX:+ShowCodeDetailsInExceptionMessages -cp /Users/evelyntoledo/Library/Application\ Support/Code/User/workspaceStorage/38641048e532be988d51b114ba605101/redhat.java/jdt_ws/Assignment\ 7_12c22a84/bin DriverAnimal

Unsorted Animal List:
[
  Name: Bob
  Breed: Siamese
  Size: 3
  ,
  Name: Chichi
  Breed: Persian
  Size: 4
  ,
  Name: Lola
  Breed: Poodle
  Size: 2
  ,
  Name: Julio
  Breed: Bulldog
  Size: 5
]

Sorting list by Name:
[
  Name: Bob
  Breed: Siamese
  Size: 3
  ,
  Name: Chichi
  Breed: Persian
  Size: 4
  ,
  Name: Julio
  Breed: Bulldog
  Size: 5
  ,
  Name: Lola
  Breed: Poodle
  Size: 2
]

```

```
PROBLEMS 3 OUTPUT DEBUG CONSOLE TERMINAL JUPYTER

Sorting by Breed:
[
  Name: Julio
  Breed: Bulldog
  Size: 5
  ,
  Name: Chichi
  Breed: Persian
  Size: 4
  ,
  Name: Lola
  Breed: Poodle
  Size: 2
  ,
  Name: Bob
  Breed: Siamese
  Size: 3
]

Sorting by Size:
[
  Name: Lola
  Breed: Poodle
  Size: 2
  ,
  Name: Bob
  Breed: Siamese
  Size: 3
  ,
  Name: Chichi
  Breed: Persian
  Size: 4
  ,
  Name: Julio
  Breed: Bulldog
  Size: 5
]
```

```
Deserialised:

Name: Lola
Breed: Poodle
Size: 2

Name: Bob
Breed: Siamese
Size: 3

Name: Chichi
Breed: Persian
Size: 4

Name: Julio
Breed: Bulldog
Size: 5
```