Exercise – Collections and Generics

Objective

This exercise looks at how to use collections and generics as well as write some more of our own simple objects. By the end of the exercise, you should be more comfortable with creating and using collections.

Overview

Setup

1. Create a class called Animal in Animal.java file
    a. This has two fields, name, and age
    b. Generate a constructor, getters, setters and a toString() method for the class
    c. Make the class abstract and add two abstract methods:
        i. sayHello() that returns a String value
        ii. move() that returns a String value
2. Create three classes called Cat, Dog and Rabbit in Cat.java, Dog.java and Rabbit.java files
    a. These class should extend Animal class
    b. Override sayHello() and move() methods in all three classes. These methods should return appropriate messages.

Creating collections

1. Create five objects from the classes you have. These should be different animal types.
2. Create an ArrayList containing all the objects
    a. Remember to give it the type parameter using <>
    b. Look at the difference when you give it the sub-type (Cat, Dog, Rabbit) rather than the supertype (Animal).
    c. Use the debugger to inspect the object as the animals are being added to it.
3. Create a HashMap of the objects
    a. First, create a HashMap that uses the name of the animal as a key and the object as the value

    b. We can also use the object itself as the key in a HashMap. Create a second HashMap that uses the object as the key and a description for value.

4. Create a set of the objects, use whichever set you like - have a look at the api documentation for this

    a. Try adding an object to the set more than once. Will it let you? Look at what is happening to the object using the debugger.

5. Print out the values in your collections. You can do this using a for loop, an enhanced for loop or the iterator. Try all three methods.

6. Find a specific object in your collections. Choose one of your objects (for example, a cat called “Bob”). Look through the API to see if there are any methods that can help you.

    Hint: For some collections you need to iterate through each item. For others you can just go directly to the object in question.

7. Now we want to sort our List. As this is the only collection that uses an ordered list of elements, we can use the sort method to do this.

    a. Sort the contents of your ArrayList by using the Collections.sort() method. This will need the animal class to extend Comparable and implement the compareTo method in either the parent class or each of the children. Sort the animals by their age.

    b. We can’t use the sort() method on a Set or a HashMap. The order of elements in a HashMap is based on the key, rather than the order they were inserted. Look at the Java API and see if there are any other types of maps or sets you can use that will sort the collections for us. Implement these collections and have a look at the output.