

Module Seven - Assignment Submission

The Assignment

Imagine your project team is developing a Java application that will serve as an educational tool to help children learn about various concepts such as vehicles, animals, sounds, etc. The application will need to draw various objects (e.g., animals, vehicles) on the video display as well as rotate objects, resize objects, and play sounds that are associated with the objects. Your project manager has specified that all objects in the application should use a common interface for drawing, rotating, resizing, and playing sounds, and that these interfaces should be reusable for other application projects as well. Some future applications will need to make sounds, some will need drawing capability, some will need rotating capability, some will need resizing capability, and some will need all capabilities. A method called `drawObject()` will be used for drawing objects, a method called `rotateObject()` will be used for rotating objects, a method called `resizeObject()` will be used for resizing objects, and a `playSound()` method will be used for playing sounds.

1. Design the interface classes `Drawable`, `Rotatable`, `Resizable`, and `Sounds`.
2. Develop a program called `Manipulator` that does the following: An `Animal` class will be used to model some basic characteristics of animals for the application. An animal will have a `name` attribute and methods to set the name and get the name. Animals can make sounds and are drawable, rotatable, and resizable. A `Vehicle` class will be used to model some basic characteristics of vehicles. A vehicle will have a `name` and an `age`, and methods to get/set these attributes. Vehicles can also make sounds and are drawable, rotatable, and resizable. Create a collection of 2 vehicles and 2 animals stored in the same array. Loop through the array and execute the `drawObject()`, `rotateObject()`, `resizeObject()`, and `playSound()` methods polymorphically for each element in the array. The `drawObject()` method should simply display the message "Drawing a Vehicle" or "Drawing an Animal"; the `rotateObject()` method should display the message "Rotating a Vehicle" or "Rotating an Animal"; and the `playSound()` method should display the message "Animal sound" or "Vehicle sound", depending upon the type of element that is in the array. The `resizeObject()` method should display "Resizing a Vehicle" or "Resizing an Animal".

Design

There isn't much room for interpretation in this assignment, so the implementation speaks to the majority of the case. The required classes and methods are defined by the specification, so please see below

Implementation

```
package module8;

// Class which tests the animal and vehicle classes
// and their implementations of interfaces by calling
// their methods while storing both types of class
// in an array
```

```
public class Manipulator {

    // Program entry point
    public static void main(String[] args) {
        // Create an array of animals and vehicles
        // and store them in the array
        Object[] objects = new Object[4];
        objects[0] = new Animal("Dog");
        objects[1] = new Vehicle("Car", 4);
        objects[2] = new Animal("Cat");
        objects[3] = new Vehicle("Bike", 2);

        // Loop through the array and call the methods
        // on the casted objects
        for (Object object : objects) {
            if (object instanceof Animal) {
                Animal animal = (Animal) object;
                animal.drawObject();
                animal.rotateObject();
                animal.resizeObject();
                animal.playSound();
            } else if (object instanceof Vehicle) {
                Vehicle vehicle = (Vehicle) object;
                vehicle.drawObject();
                vehicle.rotateObject();
                vehicle.resizeObject();
                vehicle.playSound();
            }
        }
    } // end main

} // end class Manipulator

package module8;

// Class which implements multiple interfaces
// Drawable, Rotatable, Resizable, Sounds
// for an Animal object

public class Animal implements Drawable, Rotatable, Resizable, Sounds{

    // Class Methods

    public Animal(String name) {
        this.name = name;
    }

    public void drawObject() {
        System.out.println("Drawing an Animal");
    }

    public void rotateObject() {
        System.out.println("Rotating an Animal");
    }

}
```

```
}

public void resizeObject() {
    System.out.println("Resizing an Animal");
}

public void playSound() {
    System.out.println("Animal Sound");
}

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

public String getName() {
    return name;
}

// Class Members

private String name;
} // end class Animal

package module8;

// Class which implements multiple interfaces
// Drawable, Rotatable, Resizable, Sounds
// for a Vehicle object

public class Vehicle implements Drawable, Rotatable, Resizable, Sounds {

    // Class Methods

    public Vehicle(String name, int age) {
        this.name = name;
        this.age = age;
    }

    public void drawObject() {
        System.out.println("Drawing a vehicle");
    }

    public void rotateObject() {
        System.out.println("Rotating a vehicle");
    }

    public void resizeObject() {
        System.out.println("Resizing a vehicle");
    }

    public void playSound() {
        System.out.println("Vehicle Sound");
    }
}
```

```
}

    public void setAge(int age) {
        this.age = age;
    }

    public int getAge() {
        return age;
    }

    public String getName() {
        return name;
    }

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

    // Class Members
    private int age;
    private String name;
} // end class Vehicle

package module8;

// Interface for Drawable objects

public interface Drawable {
    void drawObject();
} // end interface Drawable

package module8;

// Interface for Resizable objects

public interface Resizable {
    void resizeObject();
}

package module8;

// Interface for Rotatable objects

public interface Rotatable {
    void rotateObject();
} // end interface Rotatable

package module8;

// Interface for  objects that can play sounds

public interface Sounds {
    void playSound();
}
```

```
} // end interface Sounds
```

Output

```
Drawing an Animal  
Rotating an Animal  
Resizing an Animal  
Animal Sound  
Drawing a vehicle  
Rotating a vehicle  
Resizing a vehicle  
Vehicle Sound  
Drawing an Animal  
Rotating an Animal  
Resizing an Animal  
Animal Sound  
Drawing a vehicle  
Rotating a vehicle  
Resizing a vehicle  
Vehicle Sound
```