1. Implemente una nueva clase llamada Aerodeslizador Esta nueva clase debe implementar las interface de VehiculoTierra y VehiculoAgua con las siguientes condiciones:

a. Defina los atributos necesarios para implementar la clase de acuerdo con la lógica de los métodos que hereda esta clase. Estos atributos deben quedar con el mayor nivel de privacidad para la clase. Tome como ejemplo la clase Sedan. Defina un constructor por defecto que inicialice los atributos con valores predefinidos por usted.

b. Defina otro constructor que inicialice los atributos de esta clase con los valores pasados como parámetros a este constructor.

c. Implemente el cuerpo de los métodos que hereda esta clase teniendo en cuenta que no se deben cambiar los métodos abstractos declarados en las interfaces. Ninguna lógica de estos métodos debe ejecutar la instrucción System.out.println. Esta instrucción será exclusiva del método main de la clase ObjTaller8.

Aerodeslizador

```java
public class Aerodeslizador implements VehiculoTierra, VehiculoAgua {


    //a
    private String nombre;
    private short maxPasajeros;
    private int maxVelocidad;
    private int numeroRuedas;
    //b
    public Aerodeslizador(String nombre, short maxPasajeros, int maxVelocidad){
        this.nombre = nombre;
        this.maxPasajeros = maxPasajeros;
        this.maxVelocidad = maxVelocidad;
        this.numeroRuedas = 4;
    }
    //b
    public Aerodeslizador(String nombre, short maxPasajeros, int maxVelocidad, int
numeroRuedas) {
        this.nombre = nombre;
        this.maxPasajeros = maxPasajeros;
        this.maxVelocidad = maxVelocidad;
        this.numeroRuedas = numeroRuedas;
    }
    //b
    public Aerodeslizador() {
        this("Mi Aerodeslizador", (short) 2, 150);
    }
    //c

    public void agregarRuedas(int numeroRuedas) {
        this.numeroRuedas = numeroRuedas;
    }

    public String getNombre() {
        return this.nombre;
    }

    public int getMaxPasajeros() {
        return this.maxPasajeros;
    }

    public int getMaxVelocidad() {
        return this.maxVelocidad;
    }

    public int getNumeroRuedas() {
        return this.numeroRuedas;
    }

    public String conducir() {
        return "Conduce al vehículo: " + nombre;
    }

    String sonarClaxon() {
        return "Claxon sonando...";
    }

    public String toString() {
        return super.toString();
    }

    public String soltarAmarras() {
      return "Amarras sueltas :)";
    }
}
```

2. Implemente una nueva clase llamada Fragata Esta nueva clase debe heredar de VehiculoAgua, con las mismas condiciones indicadas en el numeral anterior

Fragata

```java
package taller8;

public class Fragata implements VehiculoAgua {

    private String nombre;
    private short maxPasajeros;
    private int maxVelocidad;

    public Fragata(String nombre, short maxPasajeros, int maxVelocidad){
        this.nombre = nombre;
        this.maxPasajeros = maxPasajeros;
        this.maxVelocidad = maxVelocidad;
    }

    public Fragata() {
        this("Mi Fragata", (short) 12, 175);
    }

    public String getNombre() {
        return this.nombre;
    }

    public int getMaxPasajeros() {
        return this.maxPasajeros;
    }

    public int getMaxVelocidad() {
        return this.maxVelocidad;
    }

    public String conducir() {
        return "Conduce al vehículo: " + nombre;
    }

    String sonarClaxon() {
        return "Claxon sonando...";
    }

    public String toString() {
        return super.toString();
    }

    public String soltarAmarras() {
      return "Amarras sueltas :)";
    }
}
```

3. Implemente una nueva clase llamada PatrullaPolicia. Esta nueva clase debe heredar de Sedan y Emergencia. Con las siguientes condiciones:

a. Implemente un constructor que inicialice los atributos heredados por la superclase

b. Implemente los atributos de acuerdo a la lógica de los métodos que está heredando esta clase.

c. Implemente el cuerpo de los métodos abstractos que está heredando esta clase. Tenga en cuenta que el método getVolumen() deben retornar el valor definido por el atributo VOLUMEN en la interface Emergencia. Tenga en cuenta las demás aclaraciones en la condición (c) del numeral (1).

d. Implemente el siguiente método dentro de la clase PatrullaPolicia: public void setVolumen() { Emergencia.VOLUMEN++;} Explique, ¿por qué esta implementación genera error?

Respuesta: El método setVolumen que se intenta definir genera un error porque está intentando modificar el valor de VOLUMEN, una constante que es declarada en la interfaz Emergencia.

e. Defina el método toString() dentro de la clase PatrullaPolicia y responda: ¿Se está sobrescribiendo el método? ¿De quién se está sobrescribiendo el método toString() en este caso?

Respuesta: Se sobrescribe de la superclase sedan.

PatrullaPolicia

```java
public class PatrullaPolicia extends Sedan implements Emergencia {
  //b
    private String nombre;
    private short maxPasajeros;
    private int maxVelocidad;
    private int numeroRuedas;

    public PatrullaPolicia() {
        this("Patrulla #01", (short) 4, 200, 4);
    }

    public PatrullaPolicia(String nombre, short maxPasajeros, int maxVelocidad) {
        this(nombre, maxPasajeros, maxVelocidad, 4);
    }

    //a
    public PatrullaPolicia(String nombre, short maxPasajeros, int maxVelocidad, int
numeroRuedas) {
        super(nombre, maxPasajeros, maxVelocidad);
        this.nombre = nombre;
        this.maxPasajeros = maxPasajeros;
        this.maxVelocidad = maxVelocidad;
        this.numeroRuedas = numeroRuedas;
    }

    //c
    @Override
    public String sonarSirena() {
        return "Sirena sonando al máximo volumen";
    }

    //c
    @Override
    public int getVolumen() {
        return VOLUMEN;
    }

    //d
    //public void setVolumen() {
      //Emergencia.VOLUMEN++;
    //}

    //e
    public String toString() {
        return super.toString();
    }
}
```

4. En el método main de la clase ObjTaller8 realice lo siguiente:

c. Cree el siguiente arreglo: ArrayList listaVehiculos = new ArrayList(); Responda: ¿Qué debe ir en el operador diamante del ArrayList?

Respuesta: ArrayList<Vehiculo> listaVehiculos = new ArrayList<>();

e. Recorra el arreglo creado con el siguiente ciclo: for (int i = 0; i < listaVehiculos.size(); i++) { //Obtiene cada elemento en el arreglo y lo almacena //en la variable v. v = listaVehiculos.get(i); //Implemente nuevo código a partir de esta línea } Responda: ¿De qué tipo debe ser la variable v en el código anterior?

Respuesta: Vehículo

g. Si se quisiera subir el volumen de la sirena de la patrulla de policía usando un método setVolumen(). Responda: ¿Qué debería hacerse y por qué?

Respuesta: No es posible modificar directamente el valor de VOLUMEN porque es una constante definida en la interfaz. En su lugar, se debe crear un atributo mutable en la clase PatrullaPolicia,

h. La línea 47 de la clase sedan está haciendo un llamado al método toString(). ¿En dónde está definido este método para su invocación.

Respuesta: Es un método heredado de la superclase Object

ObjTaller8

```
package taller8;
import java.util.*;
    public class ObjTaller8 {
      public static void main (String[] args) {
        //a
        Vehiculo s = new Sedan();
        Vehiculo a = new Aerodeslizador();
        Vehiculo f = new Fragata();
        Vehiculo p = new PatrullaPolicia("Patrulla 001",(short) 5, 200);

        //b
            if (s instanceof VehiculoTierra) {
                ((VehiculoTierra) s).agregarRuedas(4);
            }

            if (p instanceof VehiculoTierra) {
                ((VehiculoTierra) p).agregarRuedas(4);
            }

            if (a instanceof VehiculoAgua) {
                ((VehiculoTierra) a).agregarRuedas(2);
            }

            //c
        ArrayList<Vehiculo> listaVehiculos = new ArrayList<>();

        //d
        listaVehiculos.add(s);
        listaVehiculos.add(a);
        listaVehiculos.add(f);
        listaVehiculos.add(p);

        //e
        for (int i = 0; i < listaVehiculos.size(); i++) {
          Vehiculo v = listaVehiculos.get(i);

          //f
          if (v instanceof VehiculoTierra) {
             System.out.println("Nombre = " + ((VehiculoTierra)v).getNombre());
             System.out.println("Max Pasajeros = " +
((VehiculoTierra)v).getMaxPasajeros());
             System.out.println("Max Velocidad = " +
((VehiculoTierra)v).getMaxVelocidad());
             System.out.println("Max Velocidad = " +
((VehiculoTierra)v).getNumeroRuedas());
          }

          if (v instanceof VehiculoAgua) {
                System.out.println("Nombre = " + ((VehiculoAgua)v).getNombre());
                System.out.println("Max Pasajeros = " +
((VehiculoAgua)v).getMaxPasajeros());
                System.out.println("Max Velocidad = " +
((VehiculoAgua)v).getMaxVelocidad());
                System.out.println(((VehiculoAgua)v).soltarAmarras());
            }

        }


      }
}
```