

EJERCICIO 1.- SOLUCIÓN

Construir las siguientes clases:

- Clase **Vehículo**, con atributos matricula, color, numero de ruedas, cilindrada y potencia. Los dos primeros serán objetos String y los tres últimos serán enteros. La clase debe tener dos constructores parametrizados diferentes y métodos setter y getter para al menos uno de los atributos. Definir también un método toString para obtener un String con todos los datos.

```
1 public class Vehiculo {
2     protected String matricula;
3     protected String color;
4     protected int numRuedas;
5     protected int cilindrada;
6     protected int potencia;
7
8     //constructor 1
9     Vehiculo(String matricula, String color) {
10         this.matricula = matricula;
11         this.color = color;
12         numRuedas=cilindrada=potencia=0;
13     }
14
15     //constructor 2
16     Vehiculo(String matricula, String color, int numRuedas, int cilindrada, int potencia) {
17         this(matricula,color); //invoco al constructor 1 para inicializar matricula y color
18         this.numRuedas = numRuedas;
19         this.cilindrada = cilindrada;
20         this.potencia = potencia;
21     }
22
23     public int getCilindrada() {
24         return cilindrada;
25     }
26
27     public void setCilindrada(int cilindrada) {
28         this.cilindrada = cilindrada;
29     }
30
31     //Sobrescritura del método toString
32     public String toString() {
33         return "Matrícula: "+matricula+" - Color: "+color+" \nRuedas: "+ numRuedas+" - Cilindrada: "+
34             cilindrada+" - Potencia: "+potencia;
35     }
36 }
```

- Clase **Coche**, derivada de la clase Vehículo, añadiendo el atributo “numerodepuertas”, que será de tipo entero.
Debe incluir también un nuevo constructor parametrizado, el método set y get para el nuevo atributo y la sobrescritura del método toString de la clase base para añadir el dato del número de puertas.

```

1 public class Coche extends Vehiculo {
2     private int numPuertas;
3
4     public Coche(String matricula,String color,int numRuedas,int cilindrada,int potencia,int numPuertas) {
5         super(matricula, color, numRuedas, cilindrada, potencia);
6         this.numPuertas = numPuertas;
7     }
8
9     public Coche(String matricula,String color,int numPuertas) {
10        super(matricula, color);
11        this.numPuertas = numPuertas;
12        numRuedas = 4; //como es un coche, inicializo ruedas a 4
13    }
14
15    public int getNumPuertas() {
16        return numPuertas;
17    }
18
19    public void setNumPuertas(int numPuertas) {
20        this.numPuertas = numPuertas;
21    }
22
23    // Sobrescritura
24    public String toString() {
25        return super.toString()+ " - Nº puertas: "+numPuertas;
26    }
27
28 }

```

- Clase **Moto**, derivada de la clase Vehículo, añadiendo el atributo “numerodeplazas”, que será de tipo entero.
Debe incluir también un nuevo constructor parametrizado, el método set y get para el nuevo atributo y la sobrescritura del método toString de la clase base para añadir el dato del número de plazas.

```

1 public class Moto extends Vehiculo {
2     private int numPlazas;
3
4     public Moto(String matricula, String color) {
5         super(matricula, color);
6         numRuedas=2; //como es una moto, inicializo ruedas a 2
7     }
8
9     public Moto(String matricula, String color, int numRuedas,
10        int cilindrada,int potencia, int numPlazas) {
11        super(matricula, color, numRuedas, cilindrada, potencia);
12        this.numPlazas=numPlazas;
13    }
14
15    public int getNumPlazas() {
16        return numPlazas;
17    }
18
19    public void setNumPlazas(int numPlazas) {
20        this.numPlazas = numPlazas;
21    }
22
23    public String toString() {
24        return super.toString()+ " - Nº plazas: "+numPlazas;
25    }
26 }

```

- Clase con función principal, **TestVehiculo**, en la que se declaren varios objetos de la clase Coche y de la clase Moto utilizando diferentes constructores, para posteriormente ver los datos de los objetos creados a través del método sobrescrito toString.

```

1 public class TestVehiculo {
2
3     public static void main(String[] args) {
4
5         Coche cochePepe= new Coche("0000BBB", "gris plata", 4, 1800, 200, 5);
6         Coche cocheMaria= new Coche("1111CCC", "rojo", 3);
7
8         Moto motoPepe = new Moto("2222BBB", "negro");
9         //invoco al método de la clase Moto para modificar las plazas
10        motoPepe.setNumPlazas(2);
11
12        Moto motoJuan = new Moto("3333CCC", "azul", 2, 1200, 250, 1);
13
14        System.out.println("Datos del coche de Pepe:\n"+cochePepe.toString());
15        System.out.println("Datos del coche de María:\n"+cocheMaria.toString());
16        System.out.println("Datos de la moto de Pepe:\n"+motoPepe.toString());
17        System.out.println("Datos de la moto de Juan:\n"+motoJuan.toString());
18
19        //invoco a un método heredado de la superclase Vehiculo
20        cochePepe.setCilindrada(1000);
21        System.out.println("Modificamos cilindrada coche Pepe");
22        System.out.println("Datos del coche de Pepe:\n"+cochePepe.toString());
23    }
24 }
25

```

Si no introduzco número de ruedas, a los coches les pone 4 y a las motos 2

```

BlueJ: Terminal Window - ejer1
Options
Datos del coche de Pepe:
Matrícula: 0000BBB - Color: gris plata
Ruedas: 4 - Cilindrada: 1800 - Potencia: 200 - Nº puertas: 5
Datos del coche de María:
Matrícula: 1111CCC - Color: rojo
Ruedas: 4 - Cilindrada: 0 - Potencia: 0 - Nº puertas: 3
Datos de la moto de Pepe:
Matrícula: 2222BBB - Color: negro
Ruedas: 2 - Cilindrada: 0 - Potencia: 0 - Nº plazas: 2
Datos de la moto de Juan:
Matrícula: 3333CCC - Color: azul
Ruedas: 2 - Cilindrada: 1200 - Potencia: 250 - Nº plazas: 1
Modificamos cilindrada coche Pepe
Datos del coche de Pepe:
Matrícula: 0000BBB - Color: gris plata
Ruedas: 4 - Cilindrada: 1000 - Potencia: 200 - Nº puertas: 5

```