

```
1 public class Camion extends Vehiculocarga{
2     private int numEjes;
3     public Camion(String marca, String modelo, int año, int kilometraje, int capacidadcarga, int numEjes) {
4         super(marca, modelo, año, kilometraje, capacidadcarga);
5         this.numEjes = numEjes;
6     }
7     //getter y setter para Numejes
8     public int getnumEjes() {
9         return numEjes;
10    }
11
12
13    public void setnumEjes(int numEjes) {
14        this.numEjes = numEjes;
15    }
16    //sobreescribir el metodo mostrarInfo()
17    @Override
18    public void mostrarInfo() {
19        super.mostrarInfo();
20        System.out.println("Numero de Ejes:" + numEjes);
21    }
22    //sobreescribo el realizar mantenimiento
23    @Override
24    public void realizarMantenimiento(double costoPorkm, double kilometrosRecorridos) {
25        System.out.println("x: El mantenimiento del frenado en el camion.");
26        super.realizarMantenimiento(costoPorkm, kilometrosRecorridos);
27    }
28 }
29
```

C:\> Users > 50585 > OneDrive > Desktop > clase 20 de mayo 2024 > src > vehiculosPasajeros.java > vehiculosPasajeros

```

1 public class vehiculosPasajeros extends Vehiculo{
2
3     private int numPasajeros;
4
5     //constructor
6     public vehiculosPasajeros(String marca ,String modelo, int año, int kilometraje, int numPasajeros) {
7         super(marca, modelo, año, kilometraje);
8         this.numPasajeros = numPasajeros;
9     }
10
11     //getter y Setter para numPasajeros
12     public int getNumPasajeros() {
13         return numPasajeros;
14     }
15
16     public void setNumPasajeros(int numPasajeros) {
17         this.numPasajeros = numPasajeros;
18     }
19     //Sobreescribir el metodo mostrar Info()
20     @Override
21     public void mostrarInfo() {
22         super.mostrarInfo();
23         System.out.println("Numero de Pasajeros: " + numPasajeros);
24     }
25 }
26
27 }
28

```

C: > Users > 50585 > OneDrive > Desktop > clase 20 de mayo 2024 > src > Vehiculocarga.java > Vehiculocarga

```

1  public class Vehiculocarga extends Vehiculo {
2
3  private int capacidadcarga;
4
5  //constructor
6  public Vehiculocarga(String marca, String modelo, int año, int kilometraje, int capacidadcarga) {
7      super(marca, modelo, año, kilometraje);
8      this.capacidadcarga = capacidadcarga;
9  }
10 //Getter y Setter para capacidad de carga
11 public int getCapacidadcarga() {
12     return capacidadcarga;
13 }
14
15 public void setCapacidadCarga(int capacidadcarga) {
16     this.capacidadcarga = capacidadcarga;
17 }
18
19 //sobreescribir el metodo MostrarInfo
20 @Override
21 public void mostrarInfo(){
22     super.mostrarInfo();
23     System.out.println("capacidad de carga:" +capacidadcarga + " kg" );
24 }
25
26
27

```


C:\> Users > 50585 > OneDrive > Desktop > clase 20 de mayo 2024 > src > Vehiculo.java > Vehiculo > main(String[])

```

1  public class Vehiculo {
42  //metodo para mostrar la info del vehiculo
43  public void mostrarInfo() {
44      System.out.println("Marca:" +marca);
45      System.out.println("Modelo:" +modelo);
46      System.out.println("Año:" +año);
47      System.out.println("kilometraje:" +kilometraje);
48  }
49  }
50
51  //Mectodo para realizar mantenimiento basico
52  public void realizarMantenimiento(double costoPorkm, double kilometrosRecorridos) {
53      double costoTotal = costoPorkm * kilometrosRecorridos;
54      System.out.println("Costo total del mantenimiento: $" +costoTotal);
55  }
56  Run | Debug
57  public static void main(String[] args) {
58      Auto elAuto = new Auto();
59      Camion elcamion = new Camion();
60  }
61
62  }
63
64
65

```

C: > Users > 50585 > OneDrive > Desktop > clase 20 de mayo 2024 > src > Auto.java > Auto

```

1 public class Auto extends vehiculosPasajeros {
2     private String tipoCombustible;
3
4     public Auto(String marca, String modelo, int año, int kilometraje, int numPasajeros, String tipoCombustible) {
5         super(marca, modelo, año, kilometraje, numPasajeros);
6         this.tipoCombustible = tipoCombustible;
7     }
8     //getter and setter
9     public String getTipoCombustible() {
10        return tipoCombustible;
11    }
12
13    public void settipoCombustible(String tipoCombustible) {
14        this.tipoCombustible = tipoCombustible;
15    }
16    //sobreescribir el metodo mostrarInfo()
17    @Override
18    public void mostrarInfo() {
19        super.mostrarInfo();
20        System.out.println("Tipo de combustible:" + tipoCombustible);
21    }
22    //Sobreescribo el mostrar mantenimiento
23    @Override
24    public void realizarMantenimiento(double costoPorkm, double kilometrosRecorridos) {
25        super.realizarMantenimiento(costoPorkm, kilometrosRecorridos);
26        System.out.println(x:"Verificar la alineacion de las ruedas.");
27        System.out.println(x:"Inspeccionar la direccion asistida.");
28        System.out.println(x:"Lubricar componentes del sistema de direccion.");
29    }
30 }
31

```

C:\> Users > 50585 > OneDrive > Desktop > clase 20 de mayo 2024 > src > App.java > App > main(String[])

```
1 public class App {
    Run | Debug
2     public static void main(String[] args) throws Exception {
3         System.out.println(x:"Acerca del carro");
4
5         Auto auto= new Auto(marca:"Toyota", modelo:"Corrola", año:2020, kilometraje:15000, numPasajeros:5, t..."Gasolina");
6
7         Camion camion = new Camion(marca:"Volvo", modelo:"FH", año:2019, kilometraje:75000, capacidadcarga...20000, 4);
8
9         //Mostrar la informacion de cada Vehiculo
10        System.out.println(x:"Informacion del Auto:");
11        auto.mostrarInfo();
12        System.out.println(x:"Mantenimiento del sistema de direccion en el Auto.");
13        auto.realizarMantenimiento(costoPorkm:4, kilometrosRecorridos:14);
14
15        System.out.println(x:"informacion del camion:");
16        camion.mostrarInfo();
17        camion.realizarMantenimiento(costoPorkm:12, kilometrosRecorridos:30);
18    }
19 }
20
```

RESUMEN DE LOS CÓDIGOS

- En Java, el código crea objetos de las clases Auto y Camión para representar vehículos con atributos específicos y métodos para mostrar información y realizar mantenimiento. La clase Auto, heredando de VehículosPasajeros, añade tipoCombustible y métodos adicionales, sobrescribiendo para mostrar más información y hacer mantenimiento específico del sistema de dirección. La clase Camión, que extiende Vehículocarga, incluye numEjes, constructores, métodos getter y setter, y sobrescribe métodos para mostrar información adicional y realizar mantenimiento del sistema de frenado. Vehículo proporciona atributos y métodos comunes, como mostrar información y calcular costos de mantenimiento. Vehículocarga agrega capacidadcarga y métodos asociados, sobrescribiendo para mostrar esta información. VehículosPasajeros hereda de Vehículo, añade numPasajeros y sobrescribe para incluir esta información.