

Java:

```
Package com.mycompany.kernel;  
Public abstract class Vehiculo {  
    Protected String marca;  
    Protected int velocidadMaxima;  
    Public Vehiculo (String marca, int velocidadMaxima) {  
        this.marca = marca;  
        this.velocidadMaxima = velocidadMaxima;  
    }  
    Public abstract void arrancar();  
    Public void mostrarInformacion() {  
        System.out.println ("Marca: " + marca + ", Velocidad Maxima: "  
            + velocidadMaxima + " Km/h");  
    }  
    Public void detener() {  
        System.out.println ("El vehiculo se ha detenido.");  
    }  
    Public void acelerar () {int velocidad} {  
        System.out.println ("El vehiculo acelera a " + velocidad + " Km/h.");  
    }  
    Public String estado () {  
        return "El vehiculo esta en buen estado";  
    }  
}
```

## Persona.java

Package com.mycompany.Keneth;

```
public class Persona {
```

```
    private String nombre;
```

```
    public int edad;
```

```
    public persona (String nombre, int edad) {
```

```
        this.nombre = nombre;
```

```
        this.edad = edad;
```

```
}
```

```
    public void saludarPersona (String saludo) {
```

```
        System.out.println (saludo + " mi nombre es " + nombre + "
```

```
        " y tengo " + edad + " años.");
```

```
}
```

```
    public void cumpleanos () {
```

```
        edad++;
```

```
        System.out.println ("Feliz cumpleaños " + nombre + " tienes " + edad + " años");
```

```
}
```

```
}
```

## Bicicleta.java

Package com.mycompany.Keneth;

```
public class Bicicleta {
```

```
    private String marca;
```

```
    public String modelo;
```

```
    public Bicicleta (String marca, String modelo) {
```

```
        this.marca = marca;
```

```
this.modelo = modelo;
public void pedalear (int tiempo) {
    System.out.println ("La bicicleta " + marca + " ha sido pedaleada
durante " + tiempo + " minutos.");
}
public void frenar () {
    SOP ("La bicicleta " + marca + " " + modelo + " ha frenado.");
}
```

## Carro.java

```
public class Carro Extends Vehiculo {
    private String color;
    public Carro (String marca, int velocidadMaxima, String color) {
        Super (marca, velocidadMaxima);
        this.color = color;
    }
}
```

## Override

```
public void arrancar () {
    SOP ("El carro ha arrancado.");
}
public void CambiarColor (String nuevoColor) {
    this.color = nuevoColor;
    SOP ("El color del carro ha cambiado a " + color + ".");
}
public void tocarBocina () {
    SOP ("Biip! El carro esta tocando la bocina.");
}
```

## Moto.java

```
public class Moto extends Vehiculo {  
    private boolean papeles;
```

```
public Moto(String marca, int VelocidadMaxima, boolean  
            papeles) {
```

```
    super(marca, VelocidadMaxima);  
    this.papeles = papeles;  
}
```

@Override

```
public void arrancar() {
```

```
    System.out.println("La moto ha arrancado.");
```

}

```
public void hacerManiobra(String maniobra) {
```

```
    System.out.println("La moto está haciendo una maniobra: " + maniobra + ".");
```

}

```
public void hacerRuido() {
```

```
    System.out.println("Vram! La moto está haciendo ruido.");
```

}

## Kenneth.java

```
public class Kenneth {
```

```
    public static void main(String args) {
```

```
        Coche miCoche = new Coche("Toyota", 180, "Rojo");
```

```
        Moto miMoto = new Moto("Yamaha", 150, true);
```

```
        Persona miPersona = new Persona("Kenneth", 30);
```

```
        Bicicleta miBicicleta = new Bicicleta("Cannondale", "Gris");
```

```
        miCoche.modificarInformacion();
```

```
        miCoche.arrancar();
```

```
        miCoche.acelerar(120);
```

```
        miCoche.cambiarColor("Azul");
```

Mi moto. MostrarInformación();  
Mi moto. arrancar();  
Mi moto. hacermaneobra ("salto");  
Mi moto. hacerRuido();  
  
Mi persona. SaludarPersona ("Hola");  
Mi persona. CumplirAños();  
  
Mi bicicleta. Pedalear (30);  
Mi bicicleta. Frenar();

}

}

JS

app.js

class Vehiculo {

Constructor (marca, VelocidadMaxima) {

if (this.Constructor == Vehiculo) {

throw new Error ("No se puede instanciar la clase abstracta  
Vehiculo");

}

this.marca = marca;

this.VelocidadMaxima = velocidadMaxima;

arrancar () {

throw new Error ("Método 'arrancar' no implementado");

}

mostrarInformación () {

Console.log ("Marca : " + this.marca, "VelocidadMaxima : " + this.  
VelocidadMaxima + " km/h");

}

detener () {

Console.log ("El vehículo acelera y detiene.");

}

acelerar () {

Console.log ("El vehículo acelera a " + velocidad + " km/h");

}

Estado (1) y  
return "El vehículo está en 'buen estado'";

۲۴

Class carro extends Vehiculo {

Constructor (marca, velocidad máxima, color) {

Super (márcs, Velocidad máxima)

this. color = Color;

3

arrancar(), {

consola. log("El carro ha encendido");

۳

## Cambiar Color (nuevo color) {

this. Color = nuevo Color

```
console.log('El color del carro ha cambiado a ' + this.color);
```

3

Tocar Bocina ( ), {

```
console.log('Bip! El carro está tocando la bocina.');
```

Class Moto extends Vehicle {

Constructor (marca, velocidad máxima, paquetes) {

Superficie, velocidad máxima;

this. paperes = papclbs;

3

airancar() {

Censoro. log ("La moto ha arrancado");

3

hacer sonido () {

Console, log ("vram! la moto esté haciendo ruido");

3

hacer Manicura o {

Console, hoy ('la moto está haciendo una maniobra' & {maniobra}).

3

C#

Using System;

Using System.Collections.Generic;

name space EjemploAbstracto

Public abstract class Persona

    Public string nombre;

    Public int edad;

    Public bool activo;

    Public double peso;

    Public float altura;

    Public Persona(string nombre, int edad, bool activo, float altura, double peso)

{

        this.nombre = nombre;

        this.edad = edad;

        this.activo = activo;

        this.altura = altura;

        this.peso = peso;

}

    Public abstract void MostrarInformacion();

    Public abstract void Saludar();

    Public abstract void Despedirso();

    Public abstract void Dormir();

    Public abstract void Trabajar();

```
Public abstract void Caminar (String destino);  
Public abstract void Caminar (int distancia);  
Public abstract void Estudiar (String materia);  
Public abstract void Hablar (String mensaje);  
Public abstract void ActualizarDatos String nuevoNombre, int  
nuevaEdad);
```

### Clase Hija:

```
Public class Empleado & Persona  
Public string Cargo { get; set; }  
Public double Salario { get; set; }  
  
Public Empleado (String nombre, int edad, bool activo, float  
altura, double peso, String cargo, double salario) : base  
(nombre, edad, activo, altura, peso)  
{  
    Cargo = Cargo;  
    Salario = Salario;  
}  
  
Public override void MostrarInformacion ()  
{  
    Console.WriteLine ("[Empleado] " + nombre + ", Edad: " + edad + "  
    ¿ Trabajando?: " + activo + ", Cargo: " + cargo + ", Salario: " + salario);  
}
```

Public override void Saludar () => Console.WriteLine("Hola, soy " + nombre + " " + cargo + ".");

Public override void Despedirse () => Console.WriteLine(nombre + " se hasta luego!");

Public override void Dormir () => Console.WriteLine(nombre + " duerme " + horas + " horas.");

Public override void Trabajar () => Console.WriteLine(nombre + " está trabajando como " + cargo + ".");

Public override void Comer (string alimento) => Console.WriteLine(nombre + " come " + alimento + " en la oficina.");

Public override void Caminar (int distancia) => Console.WriteLine(nombre + " camina " + distancia + " metros en la empresa.");

Public override void Estudiar (String materia) => Console.WriteLine(nombre + " estudia " + materia + " para mejorar sus habilidades.");

Public override void Hablar (String mensaje) => Console.WriteLine(nombre + " dice " + mensaje);