

Java:

```
Package Com. my company. keneth;  
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 máxima: "  
            + velocidadMaxima + " Km/h");  
    }  
    public void detener () {  
        System.out.println ("El vehículo se ha detenido.");  
    }  
    public void acelerar () (int velocidad) {  
        System.out.println ("El vehículo acelera a " + velocidad + " Km/h");  
    }  
    public String estado () {  
        return "El vehículo está en buen estado";  
    }  
}
```

## Persona.java

```
Package Com.myCompany.Keneth;
```

```
Public class Persona {
```

```
    Private String nombre;  
    public int edad;
```

```
    Public person (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 cumpleaños () {
```

```
        edad ++;
```

```
        System.out.println ("Feliz cumpleaños" + nombre + " ahora 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 ("¡Bip! El carro está 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 () {
```

```
    Sop ("La moto ha arrancado.");
```

```
}
```

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

```
    Sop ("La moto está haciendo una maniobra:" + maniobra + ".");
```

```
}
```

```
Public void hacerRuido () {
```

```
    Sop ("Vrum! la moto está haciendo ruido");
```

```
}
```

```
}
```

## Keneth.java

```
Public Class Keneth {
```

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

```
        Carro miCarro = new Carro ("Toyota", 180, "Rojo");
```

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

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

```
        Bicicleta miBicicleta = new Bicicleta ("Canyon", "Gw");
```

```
        miCarro.mostrarInformación ();
```

```
        miCarro.arrancar ();
```

```
        miCarro.acelerar (120);
```

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



```

mi moto. mostrarInformación ();
mi moto. arrancar ();
mi moto. hacerManiobra ("Salto");
mi moto. hacerRuido ();

mi persona. SaludarPersona ("Hola");
mi persona. cumpleaños ();

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 + ', velocidad máxima: ' + this.velocidadMaxima + ' Km/h');
  }

  detener () {
    console.log ('El vehículo acelera y detiene.');
```

```

  }

  acelerar () {
    console.log ('El vehículo acelera a ' + this.velocidad + ' Km/h');
  }
}

```

```
estado() {  
    return "El vehiculo esta en buen estado";  
}
```

```
Class Carro extends Vehiculo {  
    Constructor (marca, velocidad maxima, color) {  
        Super (marca, velocidad maxima);  
        this.color = color;  
    }
```

```
    arrancar() {  
        Console.log("El carro ha encendido");  
    }
```

```
    Cambiar Color (nuevo Color) {  
        this.color = nuevo Color;  
        Console.log("El color del carro ha cambiado a " + this.color);  
    }
```

```
    TocarBocina() {  
        Console.log("Bip! El carro está tocando la bocina.");  
    }
```

```
Class Moto extends Vehiculo {  
    Constructor (marca, velocidad Maxima, papeles) {  
        Super (marca, velocidad Maxima);  
        this.papeles = papeles;  
    }
```

```
    arrancar() {  
        Console.log("La moto ha arrancado.");  
    }
```

```
    hacerSonido() {  
        Console.log("vrrrr! la moto está haciendo ruido.");  
    }
```

```
    hacerManiobra() {  
        Console.log("La moto está haciendo una maniobra: " + this.maniobra);  
    }
```



C#

using System;

using System.Collections.Generic;

namespace 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 MostrarInformación();

public abstract void Saludar();

public abstract void Despedirse();

public abstract void Dormir();

public abstract void Trabajar();

```
Public abstract void Comer (String alimento);  
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  
nueva Edad);
```

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 MostrarInformación ()  
    {  
        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 + "¡  
¡hasta luego!");
```

```
Public override void Dormir () => Console.WriteLine(nombre + "¡  
duerme 7 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);
```