

## Exemple de classe abstracta

### Clase AbstractoVehiculo

```
import java.awt.Color;
abstract public class AbstractoVehiculo {

    private Color ColorVehiculo;
    private byte NumRuedas;
    private short Cilindrada;
    private short Potencia;

    AbstractoVehiculo(Color color) {
        EstableceColor(color);
    }

    AbstractoVehiculo (byte NumRuedas) {
        this.NumRuedas = NumRuedas;
    }

    AbstractoVehiculo (short Cilindrada) {
        this.Cilindrada = Cilindrada;
    }

    AbstractoVehiculo(Color color, byte NumRuedas) {
        this(color);
        this.NumRuedas = NumRuedas;
    }

    AbstractoVehiculo(Color color, byte NumRuedas, short Cilindrada)
    {
        this(color,NumRuedas);
        this.Cilindrada = Cilindrada;
    }

    AbstractoVehiculo(Color color, byte NumRuedas, short Cilindrada,
        short Potencia) {
        this(color,NumRuedas,Cilindrada);
        this.Potencia = Potencia;
    }

    public void EstableceColor(Color color) {
        ColorVehiculo = color;
    }

    public Color DimeColor() {
        return ColorVehiculo;
    }

    public byte DimeNumRuedas() {
        return NumRuedas;
    }

    public short DimeCilindrada() {
        return Cilindrada;
    }

    public short DimePotencia() {
        return Potencia;
    }

    abstract public float Impuesto();
}
```

```
}
```

### Classe ACamion

```
import java.awt.Color;
public class ACamion extends AbstractoVehiculo{

    private byte NumeroDeEjes;

    ACamion(byte NumeroDeRuedas) {
        super(NumeroDeRuedas);
    }

    ACamion(Color color, byte NumeroDeRuedas) {
        super(color,NumeroDeRuedas);
    }

    ACamion(Color color, byte NumeroDeRuedas, short Cilindrada) {
        super(color,NumeroDeRuedas,Cilindrada);
    }

    ACamion(Color color, byte NumeroDeRuedas, short Cilindrada, short
Potencia) {
        super(color,NumeroDeRuedas,Cilindrada,Potencia);
    }
    ACamion(Color color, byte NumeroDeRuedas, byte NumeroDeEjes, short Cilindrada, short
Potencia) {
        super(color,NumeroDeRuedas,Cilindrada,Potencia);
        EstableceNumeroDeEjes(NumeroDeEjes);
    }

    public byte DimeNumeroDeEjes() {
        return NumeroDeEjes;
    }

    public void EstableceNumeroDeEjes(byte NumeroDeEjes) {
        this.NumeroDeEjes = NumeroDeEjes;
    }

    public float Impuesto(){
        return (super.DimeCilindrada())/30 + super.DimePotencia()*20 + super.DimeNumRuedas()*20 +
DimeNumeroDeEjes()*50);
    }

}
```

### Clase AMotocicleta

```
import java.awt.Color;
public class AMotocicleta extends AbstractoVehiculo{

    private byte NumeroDePlazas;

    AMotocicleta() {
        super((byte)2);
    }
    AMotocicleta(byte NumeroDePlazas) {
        super((byte)2);
        EstableceNumeroDePlazas(NumeroDePlazas);
    }

    AMotocicleta(Color color) {
        super(color,(byte)2);
    }

    AMotocicleta(Color color, short Cilindrada) {
        super(color,(byte)2,Cilindrada);
    }
    AMotocicleta(Color color, short Cilindrada, short Potencia) {
        super(color,(byte)2,Cilindrada,Potencia);
    }

    public byte DimeNumeroDePlazas() {
        return NumeroDePlazas;
    }

    public void EstableceNumeroDePlazas(byte NumeroDePlazas) {
        this.NumeroDePlazas = NumeroDePlazas;
    }

    public float Impuesto(){
        return (super.DimeCilindrada())/30 +super.DimePotencia()*30);
    }
}
```

### Clase APruebaVehiculos

```
import java.awt.Color;
public class APruebaVehiculos {
    public static void main(String[] args) {

        AbstractoVehiculo MotoBarata = new AMotocicleta(Color.red, (short)125, (short)25);
        AMotocicleta MotoBarata2 = new AMotocicleta(Color.red, (short)125, (short)25);
        AbstractoVehiculo MotoCara = new AMotocicleta(Color.yellow, (short)1000, (short)90);
        AbstractoVehiculo CamionNormal = new ACamion(Color.red, (byte)4, (byte)2, (short)4000,
(short)300);
        AbstractoVehiculo CamionEnorme = new ACamion(Color.red, (byte)24, (byte)6, (short)15000, (short)800);

        AMotocicleta InstanciaMotoBarata =(AMotocicleta) MotoBarata;
        InstanciaMotoBarata.EstableceNumeroDePlazas((byte)1);
        System.out.println(InstanciaMotoBarata.DimeNumeroDePlazas());

        MotoBarata2.EstableceNumeroDePlazas((byte)1);
        System.out.println(MotoBarata2.DimeNumeroDePlazas());

        System.out.println(((ACamion)CamionEnorme).DimeNumeroDeEjes());

        System.out.println(MotoBarata.Impuesto());
        System.out.println(MotoCara.Impuesto());
        System.out.println(CamionNormal.Impuesto());
        System.out.println(CamionEnorme.Impuesto());
        System.out.println(InstanciaMotoBarata.Impuesto());
    }
}
```