EJERCICIO PROGRAMACION: HERENCIA 1

Implementa las Clases siguientes:

**abstract** public class **Vehiculo** {

private String ColorVehiculo;

private byte NumRuedas;

private short Cilindrada;

private short Potencia;

Vehiculo(){};

Vehiculo(String color) {

setColorVehiculo(color);

}

Vehiculo (byte NumRuedas) {

this.NumRuedas = NumRuedas;

}

Vehiculo (short Cilindrada) {

this.Cilindrada = Cilindrada;

}

Vehiculo(String color, byte NumRuedas) {

this(color);

this.NumRuedas = NumRuedas;

}

Vehiculo(String color, byte NumRuedas, short Cilindrada) {

this(color,NumRuedas);

this.Cilindrada = Cilindrada;

}

Vehiculo(String color, byte NumRuedas, short Cilindrada,

short Potencia) {

this(color,NumRuedas,Cilindrada);

this.Potencia = Potencia;

}

public String getColorVehiculo()

{

return ColorVehiculo;

}

public byte getNumRuedas()

{

return NumRuedas;

}

public short getCilindrada()

{

return Cilindrada;

}

public short getPotencia()

{

return Potencia;

}

public void setColorVehiculo(String ColorVehiculo)

{

this.ColorVehiculo = ColorVehiculo;

}

public void setNumRuedas(byte NumRuedas)

{

this.NumRuedas = NumRuedas;

}

public void setCilindrada(short Cilindrada)

{

this.Cilindrada = Cilindrada;

}

public void setPotencia(short Potencia)

{

this.Potencia = Potencia;

}

**abstract public float Impuesto();**

}

public class **Camion extends Vehiculo** {

private byte NumeroDeEjes;

Camion(){

super();

};

Camion(byte NumeroDeRuedas) {

super(NumeroDeRuedas);

}

Camion(String color, byte NumeroDeRuedas) {

super(color,NumeroDeRuedas);

}

Camion(String color, byte NumeroDeRuedas,

short Cilindrada) {

super(color,NumeroDeRuedas,Cilindrada);

}

Camion(String color, byte NumeroDeRuedas,short Cilindrada, short Potencia) {

super(color,NumeroDeRuedas,Cilindrada,Potencia);

}

Camion(String color, byte NumeroDeRuedas, byte NumeroDeEjes, short Cilindrada, short Potencia) {

super(color,NumeroDeRuedas,Cilindrada,Potencia);

setNumeroDeEjes(NumeroDeEjes);

}

public byte getNumeroDeEjes()

{

return NumeroDeEjes;

}

public void setNumeroDeEjes(byte NumeroDeEjes)

{

this.NumeroDeEjes = NumeroDeEjes;

}

**public float Impuesto(){**

**return ((super.getCilindrada()/30 + super.getPotencia()\*20 +**

**super.getNumRuedas()\*20 + getNumeroDeEjes()\*50)\*0.01f);**

**}**

}

public class **Motocicleta extends Vehiculo** {

private byte NumeroDePlazas;

Motocicleta() {

super();

}

Motocicleta(byte NumeroDePlazas) {

super((byte)2);

setNumeroDePlazas(NumeroDePlazas);

}

Motocicleta(String color) {

super(color,(byte)2);

}

Motocicleta(String color, short Cilindrada) {

super(color,(byte)2,Cilindrada);

}

Motocicleta(String color, short Cilindrada, short Potencia) {

super(color,(byte)2,Cilindrada,Potencia);

}

Motocicleta(String color, byte NumRuedas, short Cilindrada, short Potencia, byte NumeroDePlazas)

{

super(color,NumRuedas,Cilindrada,Potencia);

setNumeroDePlazas(NumeroDePlazas);

}

public byte getNumeroDePlazas()

{

return NumeroDePlazas;

}

public void setNumeroDePlazas(byte NumeroDePlazas)

{

this.NumeroDePlazas = NumeroDePlazas;

}

**public float Impuesto(){**

**return ((super.getCilindrada()/30 + super.getPotencia()\*30)\*0.01f);**

**}**

}

Implementa una aplicación de escritorio que dé de alta Camiones y Motocicletas y los muestre. Se guardaran en listas.

static ArrayList <Camion>**listaCamiones**=new <Camion>ArrayList();

static ArrayList <Motocicleta>**listaMotocicletas**=new <Motocicleta>ArrayList();

















