

# Java

# Clases y Objetos

# Un Objeto

- Es una cosa tangible y visible
- Es una entidad de software
- Sus características son:
  - Identidad: un identificador
  - Estado: un conjunto de propiedades (atributos)
  - Comportamiento: Un conjunto de operaciones (métodos)
- Recibe también el nombre de “instancia



# Definición de Objeto



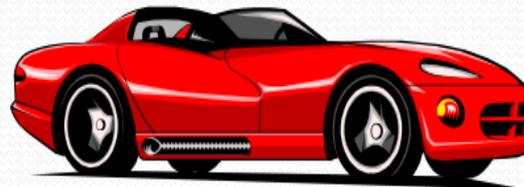
- Un objeto se caracteriza por un numero de operaciones y un estado que recuerda el efecto de operaciones.
- Un objeto tiene un estado, comportamiento e identidad.
- Un objeto es una entidad que tiene un estado y un conjunto de operaciones que operan sobre su estado.
  - El estado se representa: atributos y relaciones
  - El comportamiento se representa: por sus operaciones y métodos.

**Objeto= identidad+estado+comportamiento**

# Objeto



Automóvil



- **Atributos:**
  - color
  - velocidad
  - ruedas
  - motor

unReloj

Atributos

hora (horas, min, seg)  
dia (dia, mes, año)  
modelo  
numSerie

Métodos

getHora  
getDia  
incrementarHora  
incrementarDia  
limpiarPantalla

Operaciones de Coche  
ir, girar a la derecha, girar a la izquierda...

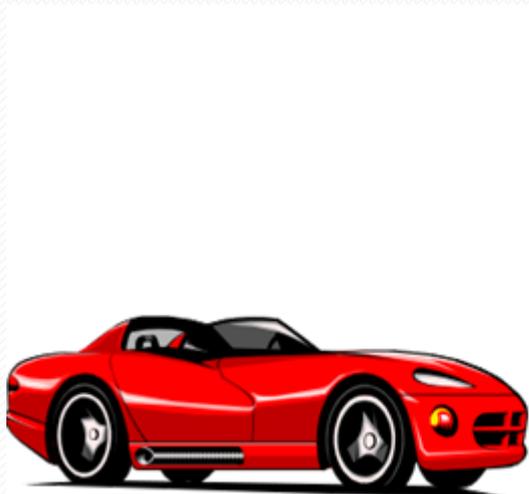


Estado de Coche  
moviéndose, parado, girando...

- **Métodos:**
  - arranca()
  - frena()
  - dobla()

# Atributo

- Es una característica fundamental de cada objeto de una clase.
- Una clase puede definir varios atributos de un objeto
- Todos los atributos tienen un valor.



## ■ Atributos:

- color
- velocidad
- ruedas
- motor

= Rojo

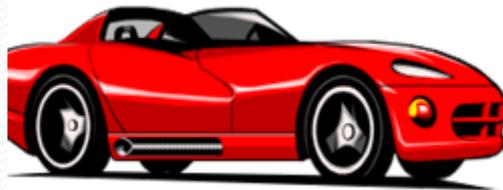
= 60 km/h

= 4

= 6 cilindros

# Método (operación)

- Es una acción que se realiza sobre un objeto para consultar o modificar su estado
- Operaciones:
  - Modificar el estado del objeto
  - Interactuar con el estado del objeto
  - Consultar el estado del objeto
  - Crear un objeto e inicializar su estado

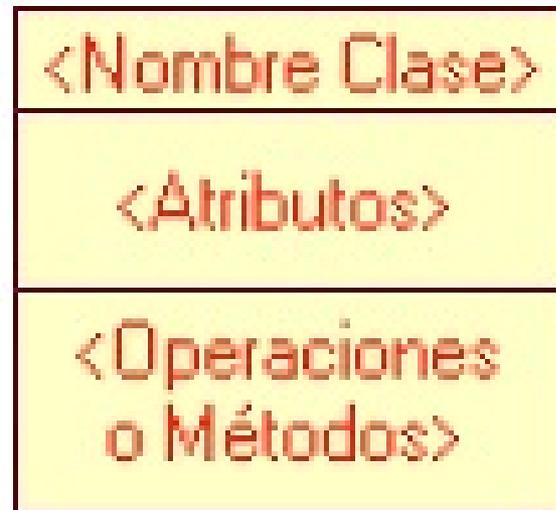


## ■ Métodos:

- arranca()
- frena()
- dobla()

# Clase

- Es la unidad básica (o estructura) que encapsula toda la información de un Objeto.
- Es un patrón para la definición de atributos y métodos para un tipo de objetos.
- A través de ella podemos modelar (una Casa, un Auto, una Cuenta Corriente, etc.).
- Una clase en UML se representa por tres divisiones:



# Clase Automóvil

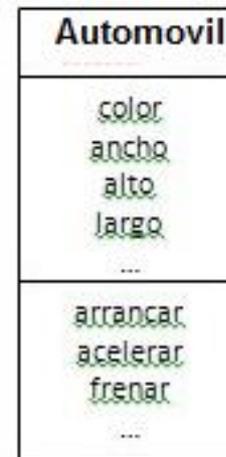
- Superior: Contiene el **nombre de la Clase**
- Intermedio: Contiene los **atributos** (o variables de instancia) que caracterizan a la Clase (pueden ser private, protected o public).
- Inferior: Contiene los **métodos u operaciones**, los cuales son la forma como interactúa el objeto con su entorno (dependiendo de la visibilidad: private, protected o public).



*Propiedades  
(atributos)*



*Métodos  
(acciones)*



# Modelo de Clases

- UML, es un Lenguaje de Modelado Unificado para representar relaciones entre elementos de un sistema de información.
- Un diagrama de clases sirve para visualizar las relaciones entre las clases que involucran el sistema.

# Entorno de Atributos: Accesos

- Los atributos o características de una Clase pueden ser de tres tipos, los que definen el grado de comunicación y visibilidad de ellos con el entorno, estos son:
  - `public (+,)`: Indica que el atributo será visible tanto dentro como fuera de la clase, es decir, es accesible desde todos lados.
  - `private (-,)`: Indica que el atributo sólo será accesible desde dentro de la clase (sólo sus métodos lo pueden acceder).
  - `protected (#,)`: Indica que el atributo no será accesible desde fuera de la clase, pero si podrá ser accesado por métodos de la clase además de las subclases que se deriven (ver herencia).

# Entorno de Métodos:Accesos

- Los métodos u operaciones de una clase son la forma en como ésta interactúa con su entorno, éstos pueden tener las características:
  - public (+,): Indica que el método será visible tanto dentro como fuera de la clase, es decir, es accesible desde todos lados.
  - private (-,): Indica que el método sólo será accesible desde dentro de la clase (sólo otros métodos de la clase lo pueden acceder).
  - protected (#,): Indica que el método no será accesible desde fuera de la clase, pero si podrá ser accesado por métodos de la clase además de métodos de las subclases que se deriven (ver herencia).

# Tipos de relaciones

- Herencia (Especialización/Generalización): 
- Agregación:
  - Agregación por Valor o Composición 
  - Agregación por Referencia 
- Asociación: 
- Dependencia o Instanciación (uso): 

# Relaciones entre Clases

- En UML, la cardinalidad de las relaciones indica el grado y nivel de dependencia, se anotan en cada extremo de la relación y éstas pueden ser:
  - **uno o muchos:** 1..\* (1..n)
  - **o o muchos:** 0..\* (0..n)
  - **número fijo:** m (m denota el número).

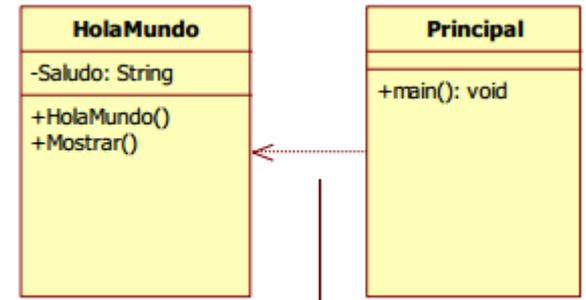
# Modelado UML

## Diagrama de Clases

```
public class HolaMundo {  
    private String Saludo;  
    public HolaMundo() {  
        Saludo="¡Hola! Mundo...";  
    }  
    public void Mostrar() {  
        System.out.println("Saludo: "+Saludo);  
    }  
}
```

```
public class Principal {  
    public static void main(String[] args){  
        HolaMundo s;  
        s=new HolaMundo();  
        s.Mostrar();  
    }  
}
```

```
Administrador: C:\Windows\system32\cmd.exe  
Microsoft Windows [Versión 6.1.7601]  
Copyright (c) 2009 Microsoft Corporation. Reservados todos los derechos.  
C:\Users\Veggeta>cd..  
C:\Users>cd..  
C:\>cd eclipse  
C:\eclipse>cd HolaMundo  
C:\eclipse\HolaMundo>cd src  
C:\eclipse\HolaMundo\src>javac Principal.java  
C:\eclipse\HolaMundo\src>java Principal  
Saludo: ¡Hola! Mundo...  
C:\eclipse\HolaMundo\src>
```



Relación de Uso

## Ejemplo 1.

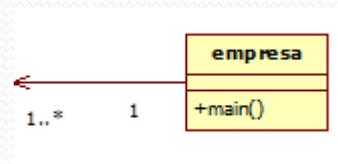
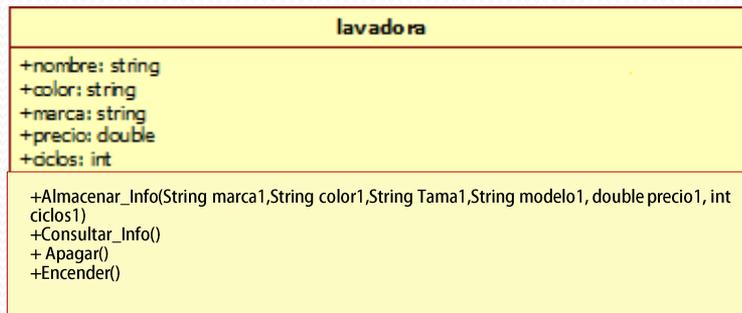
Una **empresa vende lavadoras** de diversas características como son:  
**nombre, color, marca, precio, ciclos.**

La empresa desea **almacenar información y consultar información** de cada lavadora, si se **encuentra apagada o encendida**

### 1º. Reconocer en el enunciado lo siguientes elementos:

- Sustantivo: azul- Clases
- Rojo: verbo -Acciones –Métodos u operaciones
- Verde: adjetivos calificativos –amarillo -atributos
- Naranja: herencias de Clase Padre-Clase Hija o derivada

### 2º. Dibujar su representación en UML: Diagrama de Clase y relación



### 3º. Representar el Diagrama UML en Lenguaje Java

```
public class Lavadora
```

```
{ // Atributos
```

```
    String marca;
```

```
    String color;
```

```
    String Tama;
```

```
    String modelo;
```

```
    double precio;
```

```
    int ciclos;
```

```
//constructor de la clase
```

```
    //se inicializan variables o atributos en el constructor
```

```
    public Lavadora ()
```

```
    {
```

```
    }
```

```
//posteriormente vienen los métodos
```

```
public void Almacenar_Info ( String marca1,String color1,String Tama1,String modelo1,double precio1,int  
ciclos1 )
```

```
    {
```

```
        marca = marca1;
```

```
        Tama = Tama1;
```

```
        modelo = modelo1;
```

```
        ciclos = ciclos1;
```

```
        precio = precio1;
```

```
        color = color1;
```

```
    }
```

```
public void Consultar_Info ( )
{
    System.out.println(" Marca: "+ Marca);
    System.out.println(" Color: "+ Color);
    System.out.println(" Tamaño: "+ Tam);
    System.out.println(" Modelo: "+ Modelo);
    System.out.println(" Precio: "+ Precio);
    System.out.println(" Ciclos "+ Ciclos);
}

public void Encender()
{
    System.out.println("La lavadora se encendio con un
boton");
}

public void Apagar()
{
    System.out.println("\n\nLa lavadora esta apagada");
}
} //fin clase
```

```
public class empresa
{ //esta es la clase que controla a todo
public static void main (String args[])
{
    Lavadora objLavadora = new Lavadora();
    objLavadora.almacenar_Info("easy", "grande",
"roja", "zafiro",3000.90, 5);
    objLavadora.apagar();
    objLavadora.encender();
    objLavadora.Consultar_Info();
}
```

# Identifica Clases, atributos y métodos

Laptop
-marca -No_serie -Mac -HD
+Prender() +Apagar()

Toma de corriente
-Voltaje
+Conectar_laptop() +Alimentar_Laptop()

Persona
-Sexo -Nombre -Edad
+Prender_laptop() +Apagar_laptop() +MOver_laptop() +Conectar_laptop()

Mochila
-Marca -Tamaño -Color
+Meter_laptop() +Saca_latop()

Conexion de Red
-Velocidad -Topologia -Protocolo
+Conectar_laptop()

# Ejercicio: Relaciona cada concepto

100 PUNTOS

CLASE

MÉTODO

OBJETO

ATRIBUTO

- El valor de mis atributos puede ser diferente al de los de mi semejante \_\_\_\_\_
- Yo me comporto como un plantilla \_\_\_\_\_
- A mi me gusta realizar acciones \_\_\_\_\_
- Yo puedo tener muchos métodos \_\_\_\_\_
- Yo represento el estado \_\_\_\_\_
- Yo represento el comportamiento \_\_\_\_\_
- Yo soy parte de un objeto \_\_\_\_\_
- Se puede instanciar una \_\_\_\_\_
- Yo soy una instancia o un \_\_\_\_\_

# Reflexionamos y contesta

- ¿Qué es una clase?
- ¿Qué es un objeto?
- ¿Para que se utiliza un diagrama de clases?
- ¿Qué es UML?

# Actividad en Equipo

Realiza los siguientes diagramas de clases:

- Modela la Clase Persona que tiene tres atributos: Nombre, Edad y Genero
- Modela la Clase Cliente que tiene tres atributos: No-cliente, dirección y teléfono