

ORIENTACIÓN A OBJETOS





1.

EL MUNDO DE LOS OBJETOS

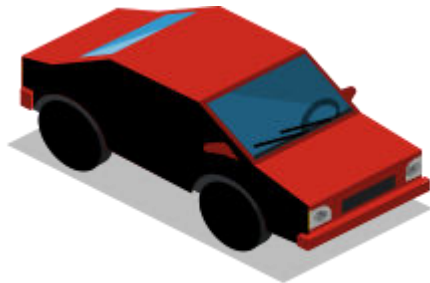
OBJETOS

- Un objeto es cualquier cosa sobre la que podemos emitir un concepto.



OBJETOS

- Podemos construir una representación de los objetos en nuestros programas.



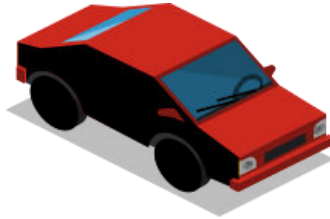
Este **automóvil de juguete** es de *plástico, rojo*, con 4 *ruedas*, 1 *volante*, que puede *moverse adelante y atrás*, ...

ESTRUCTURA Y COMPORTAMIENTO

- ▶ En general, todos los objetos tienen una estructura (como están conformado) y un comportamiento (realizan una serie de operaciones).

ESTRUCTURA

- Plástico
- 4 ruedas
- 1 volante
- ...



COMPORTAMIENTO

- Mover adelante
- Mover atrás
- ...

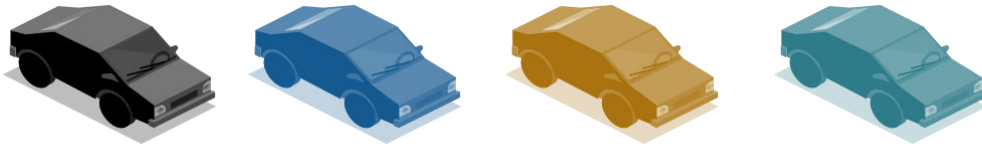
CLASE



Cada uno de estos elementos son **un objeto**. Pero todos ellos tienen algo en común; nos referimos a ellos como **GLOBO**.

CLASE

- ▶ Una clase es un molde con el que podemos construir objetos de un tipo.
- ▶ El *molde* determina las características y el comportamiento que podrá tener ese objeto.
- ▶ A ese molde, como tal, no lo consideramos un objeto.



CLASE vs. OBJETOS

ESTRUCTURA

- Nombre
- Edad
- Color piel
- Profesión
- Estado civil



COMPORTAMIENTO

- Hablar
- Caminar
- Mirar
- Nacer
- Morir





2.

PROGRAMACIÓN ORIENTADA A OBJETOS

PARADIGMA

- ▶ Paradigma significa modelo.
- ▶ Es la forma en la que se entiende que hay que estructurar un programa.
- ▶ Existen múltiples paradigmas.



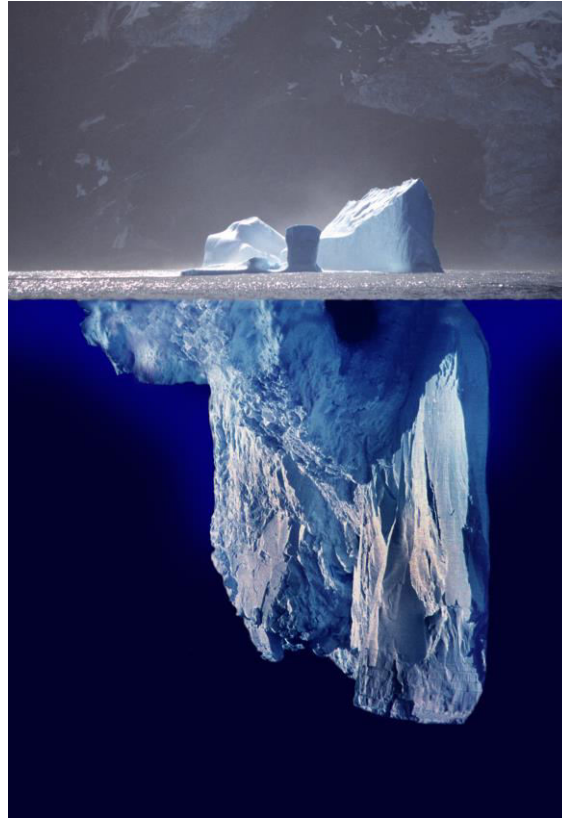
PROGRAMACIÓN ORIENTADA A OBJETOS

- ▶ Estructura todas las partes de un programa mediante objetos.
- ▶ Los objetos interaccionan entre ellos mediante un **paso de mensajes**.



ENCAPSULACIÓN

- Los objetos conocen solamente su estructura, no la de los demás.



PROGRAMACIÓN ORIENTADA A OBJETOS EN JAVA

- ▶ Java es totalmente orientado a objetos.
- ▶ Nos permite manejar clases y objetos.
- ▶ Palabra reservada **class**
- ▶ La declaración e implementación de una clase estará en un mismo fichero.
- ▶ Los nombres de las clases usan notación *UpperCamelCase*.

DEFINICIÓN E IMPLEMENTACIÓN DE UNA CLASE

```
<modificador> class NombreDeLaClase {
```

```
    //propiedades  
    int propiedad1;  
    String propiedad2;  
    float propiedad3;  
    //...
```

```
    //metodos  
    void metodo1() {  
        //...  
    }
```

```
    //...
```

```
}
```

INSTANCIACIÓN DE OBJETOS DE UNA CLASE

- ▶ Construimos objetos con el *molde* de la clase.
- ▶ Sintaxis parecida a la declaración de una variable de tipo primitivo.
- ▶ Uso del operador **new**.

Persona persona = new Persona();

Tipo (clase)

Nombre del objeto (referencia)

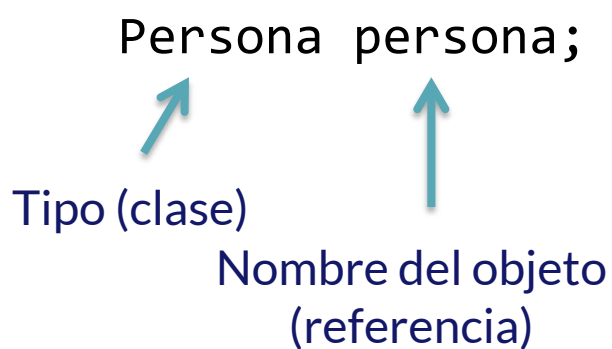
Operador de instanciación

Constructor

VALOR NULL

- ▶ NULL significa ausencia de información.
- ▶ Palabra reservada **null**, para comparar.
- ▶ Podemos declarar una referencia a un tipo de objeto, pero no construir ninguno.

Persona persona;



Tipo (clase)

Nombre del objeto
(referencia)

En este caso, esta referencia ahora mismo no nos permite acceder a ningún objeto, y almacena un valor NULL.

INTERACCIÓN ENTRE OBJETOS

- El paso de mensajes se realiza llamando a los *métodos* de un objeto desde otro.

```
persona.nacer();  
persona.hablar();  
persona.caminar();  
persona.morir();
```

MODIFICADORES DE ACCESO

- ▶ Nos permiten indicar quien puede hacer uso de una clase, o de sus atributos y métodos.
- ▶ *public*: cualquiera
- ▶ *private*: solo la propia clase
- ▶ *protected*: la propia clase y sus derivados
- ▶ Por defecto: las clases cercanas (que estén en el mismo paquete).

PAQUETE

- ▶ Es una unidad organizativa, que puede contener una o más clases.
- ▶ A nivel práctico, es un directorio (o un árbol de directorios).
- ▶ Nos permiten organizar las clases de forma lógica.
- ▶ Indicamos que una clase pertenece a un paquete en la cabecera de la misma (*package*).