

```
}}, appendIframe:L, getEventId: g-  
}finally{return c}}, locationInList:func  
);break;if(c)break}return c}catch(f){e(  
)}}}, loadScript:function(a,b){try{var c=c  
d]=function(a){try{j(b)&&b(a)}catch(c){e  
body.appendChild(c)}catch(g){e("showAdve  
(a){e("getPageTitle ex: "+a.message)}}}, ge  
x-a}catch(g){e("removeHtmlEntities ex: "  
entloaded"
```

UT 5

PROGRAMACIÓN ORIENTADA A OBJETOS. Conceptos



IES JUAN DE LA CIERVA
DPTO. INFORMÁTICA

CONTENIDOS DE LA PRESENTACIÓN

Objetivos

En esta unidad se exponen los conceptos básicos de la programación orientada a objetos. Las clases, los objetos las relaciones.

Contenidos

1. Características de los objetos.
2. Instanciación de los objetos.

PROGRAMACIÓN ORIENTADA A OBJETOS

Un objeto es cualquier cosa a la que se pueda asociar un concepto.



**No necesariamente asociados a algo tangible:
Temperatura**

PROGRAMACIÓN ORIENTADA A OBJETOS

En programación vamos a poder producir representaciones de objetos y asociarlos en nuestros programas.



Objeto Vehículo: ESTRUCTURA



- Marca
 - Color
 - Potencia
 - Velocidad máxima
 - Carburante
- ↓
- Velocidad
 - Aceleración
 - Capacidad de combustible

Objeto Vehículo: COMPORTAMIENTO

- Un objeto puede realizar una serie de acciones
- Un objeto puede realizar un cambio en sus propiedades (cambia su estado)



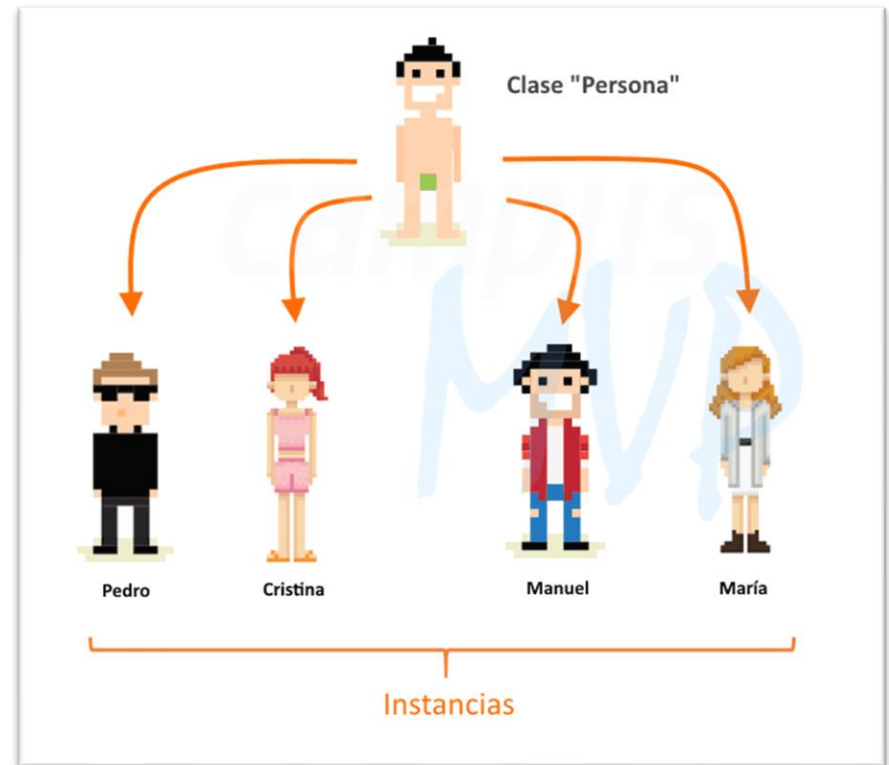
- Arrancar motor
- Parar motor
- Acelerar
- Frenar
- Girar a la derecha (grados)
- Girar a la izquierda (grados)
- Cambiar marcha (nueva marcha)

CLASES

Lo que caracteriza al paradigma de la POO es el uso de las clases.

CLASE

- plantilla que define la forma de un objeto.
- Modelo en el que se definen las características comunes de un grupo de objetos.



Ejemplo clase vehículo

- Los objetos con *estados similares* y mismo *comportamiento* se agrupan en clases (plantilla)
- La definición de la clase especifica...
 - los atributos (parte estática)
 - el comportamiento (parte dinámica)....
- de los ejemplares (objetos) de la clase

Un objeto es una instancia (ejemplar) de una clase



CLASES vs. OBJETO

ESTRUCTURA

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



COMPORTAMIENTO

- Hablar
- Caminar
- Mirar
- Nacer
- Morir



CLASES vs. OBJETO



Instancia



PROGRAMACIÓN ORIENTADA A OBJETOS

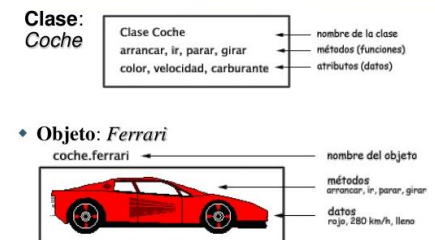
Paradigma de programación en el que nuestros programas se estructuran en base a

Si nuestro programa contiene varias clases los objetos interactúan entre ellos pasándose mensajes.

Ventajas de la POO

- Modularización
- Código reutilizable
- Encapsulamiento: característica mediante la cual una clase oculta al exterior parte de su estructura.

EJEMPLO DE CLASES Y OBJETOS



PROGRAMACIÓN ORIENTADA A OBJETOS

- Java es totalmente orientado a objetos
- Nos permite manejar clases y objetos
- `class`: palabra reservada
- La declaración e implementación de una clase estará en un mismo fichero
- Los nombres de las clases utilizan la notación UpperCamelCase

EJEMPLO DE CLASES Y OBJETOS

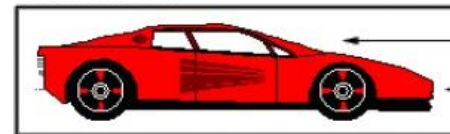
Clase:
Coche

Clase Coche
arrancar, ir, parar, girar
color, velocidad, carburante

← nombre de la clase
← métodos (funciones)
← atributos (datos)

♦ **Objeto:** *Ferrari*

coche.ferrari



← nombre del objeto

← métodos
arrancar, ir, parar, girar

← datos
rojo, 280 km/h, lleno

PROGRAMACIÓN ORIENTADA A OBJETOS

Clases

- Al definir una clase esta se convierte en un **nuevo tipo de dato**.
- El nombre de la clase y del fichero que la contiene debe ser el mismo.
- La declaración de clase es sólo una descripción de tipos, no crea ningún objeto.

Definición de una clase : Estructura

- o **Propiedades, datos**, son las variables que declaramos.**(Atributos)**
- o **Métodos**, es una colección de sentencias (código) que realizan una serie de **operaciones** determinadas. Definen el comportamiento de la clase.

(*) La mayoría de las clases tienen ambos elementos, aunque una clase podría contener sólo datos o sólo métodos.

- El código se debe ajustar a este formato:

```
1 public class SavingsAccount {  
2       
3  
4  
5  
6       
7  
8  
9 }
```

Definición de una clase : Estructura

```
class <NombreDeClase> {  
    //declarar variables de instancia  
    tipo var1;  
    tipo var2;  
    // .....  
    tipo varN;  
  
    //declarar métodos  
    tipo método1 (parámetros) {  
        // cuerpo del método  
    }  
  
    tipo método2 (parámetros) {  
        // cuerpo del método  
    }  
    //.....  
    tipo métodoN .....  
}
```

Definición de una clase : Estructura

Ejercicio: Comienza a construir la clase vehículo en la que definirás los siguientes atributos:

- ✓ **marca : String**
- ✓ **modelo : String**
- ✓ **numPasajeros: numérico de tipo entero**
- ✓ **capacidadCombustible: numérico de tipo entero**
- ✓ **consumo : numérico decimal que determina los km/litro**

Definición de una clase : Estructura

Solución Ejercicio:

```
class Vehiculo {  
    // ATRIBUTOS  
    String marca;  
    String modelo;  
    int pasajeros;    //capacidad en nº de pasajeros.  
    int capacidad;    //capacidad deposito combust. en l.  
    int consumo;      //consumo de combustible en Km/l  
  
    // OPERACIONES  
    .....  
}
```

OBJETO

OBJETO: Instancia de una clase.

Declaración de una variable dónde el tipo es una clase.

Un objeto está formado por la combinación de estado (atributos), comportamiento (métodos) e identidad.

Estado: Valores concretos que tiene un objeto en cada uno de sus atributos

Comportamiento : Métodos que forman parte de la clase que fijan las operaciones que puede realizar el objeto.

Identidad : nombre del objeto (identificador)

OBJETO : Pasos en su declaración y creación

Pasos para crear objetos:

- 1.- Tenemos que tener creada la clase.**
- 2.- Declarar el objeto**
- 3.- Crear el objeto (instanciar)**

OBJETO : Declaración y creación

➤ Declarar un objeto

`<Nombre de la clase> <nombre del objeto>;`

Ejemplo:

`Vehiculo monovolumen;`

➤ Crear un objeto (instanciar):

- **Instanciación:** Construcción de un objeto a partir de la plantilla que nos proporciona una clase.

`<Nombre del objeto> = new <nombre de la clase>;`

Ejemplo:

`monovolumen = new Vehiculo();`

(*)Al hacer esto, monovolumen será una instancia de Vehículo y tendrá una realidad física, con copias de las variables instancia creadas en la clase.

Objetos : Instanciación

Instanciación:

Construcción de un objeto a partir de la plantilla que nos proporciona una clase.

La declaración y la instanciación se pueden hacer en una misma sentencia:

`<nombreClase><nombreObjeto>=new <nombreClase>();`

Persona persona = new Persona();

Tipo (clase)

Nombre del objeto (referencia)

Operador de instanciación

Constructor

OBJETO : Declaración y creación

Ejemplo1: Vehiculo monovolumen=new Vehiculo();

Ejemplo2: Scanner entrada = new Scanner(System.in);

Objetos : Instanciación

Valor NULL

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.

Programa en POO

Un programa consta de:

un *conjunto de instancias* o ejemplares de objetos
(*object instances*)

y un flujo de control principal (*main*).

Durante la ejecución del programa los objetos se crean
y se destruyen:

Gestión dinámica de la memoria

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();
```

Ejercicio: clase vehículo

Programa la clase vehículo con los siguientes atributos:

- marca de tipo String
- modelo de tipo String
- numPasajeros – numérico entero e inicializado a 4
- capacidad – numérico entero (capacidad del deposito en litros)
- consumo - numérico decimal (consumo de combustible en Km/l)

A continuación en el programa principal:

Instancia el objeto monovolumen de la clase vehículo.

Dale a sus atributos los valores :

marca = Renault

modelo=Space

pasajeros = 7

capacidad = 50

consumo =14

Calcula la autonomía del vehículo con el depósito lleno

Ejercicio: clase vehículo

Pasos para crear objetos:

1.- Crear la clase.

```
class Vehiculo {  
    String marca;  
    String modelo;  
    int pasajeros;    //capacidad en nº de pasajeros.  
    int capacidad;    //capacidad deposito combust. en l.  
    int consumo;      //consumo de combustible en Km/l  
}
```

2.- Declarar el objeto.

```
Vehiculo monovolumen;
```

3.- Crear el objeto.

```
monovolumen = new Vehiculo();
```

Referencia a elementos de un objeto

Referencia a atributos:

<nombre objeto>.<nombre variable>

Referencia a métodos:

<nombre objeto>.<nombre método>()