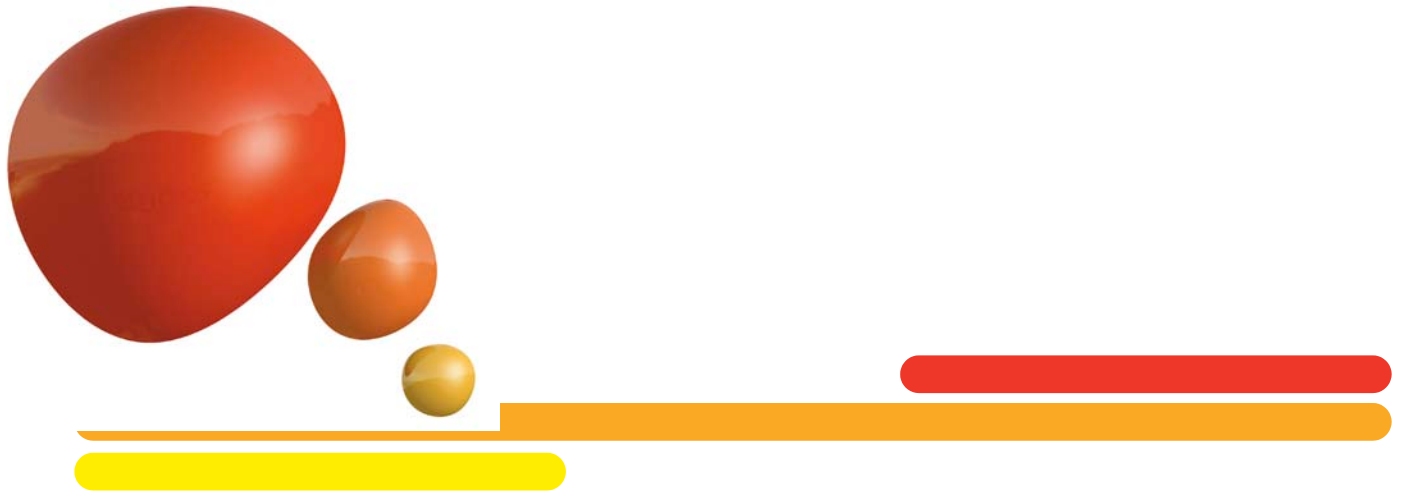
The background consists of three overlapping rounded rectangles: a yellow one on the left containing a white thought bubble, an orange one in the middle, and a red one on the right.

LA PROGRAMACIÓN ORIENTADA A OBJETOS



ÍNDICE

LA PROGRAMACIÓN ORIENTADA A OBJETOS

1. OBJETOS	3
2. CLASES	4



La programación orientada a objetos

1. OBJETOS

Características de los Objetos

Los objetos están constituidos por propiedades o atributos y métodos.

Los objetos están organizados en las clases que representan las características comunes a todos los objetos que pertenecen a ellas.

Ejemplos de los Objetos

Revisemos lo anterior, a través de un ejemplo.



La clase coche tiene las propiedades color y motor y los métodos arrancar() y frenar(). Por lo tanto, todos los objetos que pertenezcan a la clase coche, llevarán implícitas estas propiedades así como el acceso a sus métodos.

Dentro de la clase coche, las propiedades color y motor no contienen ningún valor, simplemente se declaran, sin embargo, cuando creamos un objeto de esa clase es cuando los valores de las propiedades toman un valor acorde con el tipo que tienen asignado.

Características de la POO

Las características que hacen que un lenguaje sea o se considere orientado a objetos son:

Herencia:

Consiste en posibilidad de crear clases que hereden todos los miembros (atributos y métodos) definidos en clases ya existentes. Permite reutilizar enormes cantidades de código.

La programación orientada a objetos

Polimorfismo:

Permite que una instrucción o grupo de instrucciones puedan ser utilizadas con diferentes tipos de objetos. Permite reutilizar enormes cantidades de código.

Encapsulación:

Permite proteger el acceso a los miembros de la clase para evitar que puedan ser utilizados desde el exterior. Existen diferentes niveles de visibilidad como: private, default, protected, public.

Los objetos son modelos de programación que comprenden una serie de propiedades por medio de variables, y una funcionalidad por medio de métodos.



2. CLASES

Las clases son la generalización de los objetos, de forma que un conjunto de objetos con las mismas características y funcionalidad que formarán una clase.

Surge ahora el término de **instancia**. Una instancia de una clase no es más que un objeto particular de esa clase. Cuando creamos un objeto de una determinada clase, diremos que ese objeto se ha instanciado de la clase a la cual pertenece.

Clases primitivas

Se encuentran organizadas en paquetes. si queremos utilizar una clase(primitiva o creada por nosotros) en un programa Java debemos importarla mediante la sentencia import.

Ejemplo:

<code>import Java.applet.Applet.;</code>		importamos la clase Applet del paquete Java.applet
<code>import Java.awt.Color;</code>		importamos la clase Color del paquete Java.awt

La programación orientada a objetos

Si nos interesara incluir todas las clases del paquete Java.awt, teclearemos el comando import seguido del nombre del paquete que nos interese cargar y después el símbolo * (asterisco) para hacer referencia a todas las clases incluidas en ese paquete.

Ejemplo:

`import Java.awt.*;` → importamos todas las clases del paquete Java.awt

Parámetros de una clase

Revisemos ahora todos los parámetros que pueden aparecer a la hora de definir una clase: Definición de una clase:

```
[Modificador] class Nombre_clase [extends Nombre_clase] [implements Nombre_Interface]
{
}
```

[Modificador]: aplicable a clases, se verán más adelante los posibles modificadores

class: palabra reservada. Sirve para declarar una clase. Todos los atributos y métodos de la clase se definirán dentro de las llaves de abajo.

Nombre de Clase: es el nombre que le damos a la clase.

[extends Nombre_clase]: indica que la clase hereda de otra clase.

[implements Nombre_interface]: indica que se implementa un interface Java.