



Programación Orientada a Objetos

Se trata de un paradigma de programación donde se organiza el código en unidades denominadas clases, de las cuales se crean objetos que se relacionan entre sí, simplificando el desarrollo de software y el mantenimiento de las aplicaciones

P00 vs Programación estructurada

Programación Orientada a Objetos	Programación Estructurada
Forma de programar más cercana a cómo expresarías las cosas en la vida real	Pretende resolver un problema de principio a fin en una sola estructura de código
Análisis orientado a los objetos	Análisis orientado a los procesos del sistema
Existen diferentes formas de afrontar la solución a un problema	Técnicas estructuradas básicas: Secuencia, selección, repetición
Las funciones y los datos son encapsulados en una entidad	Las funciones y los datos son manejados como entidades separadas

Clases y Objetos

// Clases

Grupo de variables y operaciones agrupadas en módulos coherentes. Una clase puede tener atributos, constructores y métodos. Podría decirse que es una plantilla que define la forma de un objeto.

```
public class Libro {
    String nombre;
    String autor;
}
```

```
public class NombreClase {
    //Declaración variables de instancia
    tipoDato variable1;
    tipoDato variable2;

    //Declaración de métodos
    ...
}
```

// Objetos

Son instancias de una clase, es decir, cuando creas un objeto, este pertenece a cierta clase y existe una representación física de esa clase en memoria.

```
Libro libro = new Libro();
```

Definimos el tipo de dato, es decir la clase a la que va a pertenecer el objeto

Utilizamos la palabra clave **new**

Indicamos un nombre significativo para la variable

Llamamos al constructor de la clase

¿No sabes a qué nos referimos? ¡No te preocupes! Lo veremos más adelante

Métodos

Conjunto de instrucciones definidas dentro de una clase que realizan una tarea determinada.

¿Cómo se escribe un método?

Se identifican por tener paréntesis después de su nombre. Un método tendrá:

- Un modificador de acceso
- Un tipo de retorno
- El nombre (debe ser descriptivo)
- Puede o no tener parámetros.

Modificador de Acceso

El método puede ser accedido desde cualquier parte del programa

Tipo de retorno

Se especifica el tipo de dato que retorna el método

Parámetros

Son variables que reciben el valor de los argumentos pasados al método, si el método no tiene parámetros la lista estará vacía

```

public int sumarNumero(int numero1, int numero2) {
    //Cuerpo del método
}

```

Nombre

Debe ser un nombre descriptivo a lo que realiza el método, puede ser cualquier identificador válido que no haya sido utilizado en el contexto actual

Método main()

Los programas en Java inician su ejecución mediante el método `main()`, en otras palabras, es el punto de entrada de la aplicación.

La palabra reservada **static** indica que no tiene que crearse un objeto de esta clase para llamar a dicho método

Este método no retorna ningún valor, este tipo de retorno se llama **void**

El nombre del método debe ser **main**

Debe aceptar un **array de String**. El nombre de los parámetros o argumentos del método puede ser cualquier identificador válido

La palabra reservada **public** indica que puede ser accedido desde cualquier parte del programa

```

public class Main {
    public static void main(String[] args) {
        System.out.println("Hola Mundo");
    }
}

```

Métodos de instancia vs Métodos de Clase

Los **métodos de instancia** pueden ser accedidos mediante una instancia u objeto de una clase.

```

public class Libro {
    private String nombreLibro;
    private String autor;

    public String mostrarNombreConAutor() {
        System.out.println(nombreLibro + " " + autor);
    }

    public String mostrarNombreSinAutor() {
        System.out.println(nombreLibro);
    }
}

//Sintaxis de llamada:
//instancia.método(parámetros)

Libro libro = new Libro();
libro.mostrarNombreConAutor();
libro.mostrarNombreSinAutor();

```

Métodos de clase

Un **método de clase** es aquel que puede ser llamado sin existir una instancia de la clase. Un ejemplo típico de uso es la clase `java.lang.Math`:

```

package java.lang;
public class Math {
    //Devuelve el menor valor de dos valores int.
    public static int min(int a, int b) {...}

    //Devuelve el mayor valor de dos valores int.
    public static int max(int a, int b) {...}

    //Devuelve el valor del primer argumento elevado a la potencia del segundo argumento.
    public static double pow(double a, double b) {...}
}

//Sintaxis de llamada:
//Clase.método(parámetros)

int menor = Math.min(2,3);
int mayor = Math.max(5,8);
double potencia = Math.pow(2,2);

```

Constructores

Toda clase en Java tiene un constructor, aunque no lo escribamos. Si no lo haces, Java creará un **Constructor por defecto** sin parámetros. El constructor se encarga de inicializar el valor de cada atributo de la nueva instancia.

Los atributos de tipo primitivo se inicializan a **0 o false** por defecto, mientras que los atributos de tipo objeto (referencia) se inicializan en **null**.

Java no permite que las variables miembro de una nueva instancia queden sin inicializar, por lo que si las mismas no se inicializan apuntan a null.

```
Libro libro = new Libro();
```

En este caso **Libro()** es un método ya que está seguido de paréntesis. Se trata de un **tipo especial de método** llamado constructor, el cual nos permite crear un nuevo objeto.

Declaración de un constructor

- El nombre del constructor debe coincidir con el nombre de la clase
- No tiene tipo de retorno
- La existencia de parámetros es opcional
- Puede existir más de uno, aunque sólo uno se ejecutará al momento de crear el objeto

```
//Constructor sin parámetros  
public Libro() {  
}
```

```
//Constructor con parámetros  
public Libro(String nombre, String autor) {  
    this.nombre = nombre;  
    this.autor = autor;  
}
```

Sobrecarga de un constructor (Overloading)

La sobrecarga permite que puedan declararse *varios constructores* para una misma clase (debe tener el mismo nombre que la clase), siempre y cuando, tengan un tipo y/o número de parámetros distinto. Hacerlo permite construir un objeto de diferentes maneras.