

# Programación Java

Tutorial Java. Aprende a programar con Java desde cero.

 Buscar

Última entrada Teoría Ejercicios Ejercicios POO C++

## Programación procedimental vs programación orientada a objetos

La Programación Orientada a Objetos (POO o también OOP del inglés Object Oriented Programming) es el paradigma de programación (modelo de programación) más utilizado en la actualidad para el desarrollo de software. Surge en los años 70 del siglo XX y fue en la década de los 90 donde se popularizó su uso.

Antes de la aparición del modelo de programación orientado a objetos se utilizaba el modelo imperativo, que es el modelo que hemos seguido durante el curso hasta ahora, en el que se definen por un lado los datos y por otro lado las operaciones que vamos a realizar con esos datos. Datos y operaciones son cosas independientes.

La idea principal de la POO es el concepto de objeto que se puede entender como una entidad formada por datos y las operaciones a realizar con esos datos. En este caso los datos y las operaciones no son independientes.

Para ver las diferencias entre la programación clásica y la POO, comenzaremos con un repaso a los conceptos de programación imperativa, estructurada y procedimental para seguir después con los conceptos básicos de la POO.

### PROGRAMACIÓN IMPERATIVA, ESTRUCTURADA Y PROCEDIMENTAL

El paradigma de programación más antiguo es el **modelo imperativo**.

En este modelo los programas están formados por una serie de instrucciones que describen paso a paso y de forma precisa los pasos a seguir para resolver un problema.

Las instrucciones se ejecutan unas a continuación de otras en el orden en que han sido escritas.

En sus orígenes cuando había que romper el orden secuencial de ejecución de las instrucciones para realizar un bucle o una instrucción condicional, se utilizaban saltos (instrucción *goto*). Las instrucciones de salto hacían que los programas fuesen complicados de seguir y depurar.

```
x = 0
etiqueta1:
x = x + 1
print x
if x < 10
    goto etiqueta1
```

El paradigma de programación estructurada es un modelo de programación basado en el modelo imperativo que usa las estructuras de control: estructuras condicionales, bucles while, for, etc. Este modelo evita las instrucciones de salto (goto) con lo que se consigue un código más sencillo de leer y de menor complejidad.

```
x = 0
do{
    x = x + 1
    print x
}while(x < 10)
```

El paradigma de programación procedimental, también llamado programación modular o programación funcional, es un modelo de programación que utiliza el modelo de programación estructurada y se basa en dividir las tareas que debe hacer un programa en partes más pequeñas. Estas partes en las que se divide el trabajo a realizar se llaman procedimientos aunque también se les conocen como métodos, funciones, subrutinas, etc.

Un procedimiento (un método en lenguaje Java) contiene una serie de instrucciones que realizan una tarea muy concreta.

### PUBLICACIONES DEL BLOG



JAVA - Ejercicios básicos resueltos



### ENTRADAS POPULARES

En este modelo los programas están formados por una serie de datos y una serie de métodos que se van llamando en un determinado orden para resolver el problema.

En el siguiente diagrama puedes ver la relación entre los paradigmas imperativo, estructurado y procedimental:



Todos los programas que hemos realizado hasta ahora durante el curso han seguido el modelo procedimental.

En la programación procedimental:

- Los datos y las acciones a realizar sobre ellos son cosas distintas.
- Se definen por un lado los datos y se definen por separado una serie de métodos o procedimientos que operan sobre ellas.

En el siguiente ejemplo se puede ver el estilo de los programas procedimentales:

```
public static void main(String[] args) {
    String nombre="Arsenio", apellido1="Leante", apellido2="Gomez";
    int edad = 30;
    String nombre2;

    nombre2 = nombreCompleto (nombre, apellido1, apellido2);

    if (esMayorEdad (edad)){
        System.out.println (nombre2 + " es mayor de edad");
    }else{
        System.out.println (nombre2 + " no es mayor de edad");
    }
}

public static String nombreCompleto (String nom, String apel1, String ape2){
    return nom + " " + apel1 + " " + ape2;
}

public static boolean esMayorEdad(int n){
    if (n>=18)
        return true;
    else
        return false;
}
```

Se crean los datos

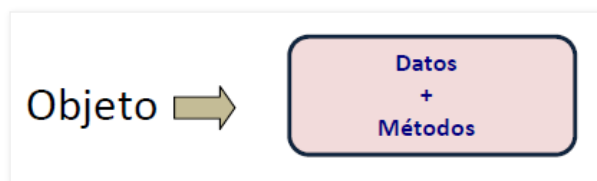
Los datos se envían a los métodos para que operen con ellos y devuelvan resultados

## PROGRAMACIÓN ORIENTADA A OBJETOS

En el **paradigma de programación orientada a objetos** los programas se organizan de forma similar a la organización de los objetos en el mundo real.

En POO cuando definimos los datos también definimos las acciones a realizar sobre ellos.

Podemos definir un **objeto** como un conjunto de **datos + métodos**.



Los métodos definen las operaciones que se pueden realizar con los datos del objeto.

Podemos definir la programación orientada a objetos como un paradigma de programación (una forma de programar) en el que el código se organiza en unidades llamadas clases de las que se crean objetos que interactúan entre ellos para conseguir resolver el problema.

La principal diferencia entre el modelo procedimental y el orientado a objetos es que en el modelo procedimental los datos y las operaciones que se realizan sobre ellos se definen por separado. En la POO los datos llevan asociados una serie de operaciones a realizar.

El ejemplo anterior escrito ahora utilizando orientación a objetos en Java sería así:

1. Se declara la clase Persona que contiene los datos de la persona y las operaciones (métodos) a realizar con esos datos:



**Ejercicios Básicos Resueltos 1**  
Relación Nº 1:  
Ejercicios 1, 2 y 3  
Empezaremos por unos ejercicios básicos de programas Java con estructura secuencial, es decir, en es...

**Calcular el factorial de un número en Java**

Factorial en java Programa que calcule el factorial de un número entero que se introduce por teclado. El factorial de un número se expresa m...



**Java printf para dar formato a los datos de salida**

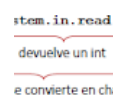
Vamos a ver como utilizar printf para dar formato a los datos que se imprimen por pantalla en Java. Este problema se nos plantea por ejempl...

**Ejercicio Básico POO Java. Clase Cuenta**

Ejercicio básico de programación orientada a objetos en Java Escribe una clase Cuenta para representar una cuenta bancaria. Los datos de l...



Estructuras de control en Java  
Las estructuras de control determinan la secuencia de ejecución de las sentencias de un programa. Los programas contienen instrucciones...



**Leer un char por teclado en Java**  
Cómo leer un carácter desde teclado en Java La clase Scanner NO CONTIENE un método nextChar() para leer un dato de tipo char desde tecl...



Programación Java  
Enrique García Hernández

## SEGUIDORES

## TRANSLATE

Seleccionar idioma ▼

## LENGUAJE C++

**Programacion C++**  
Números amigos en C++

```

public class Persona {
    //Datos
    private String nombre;
    private String apellido1;
    private String apellido2;
    private int edad;

    //Métodos
    public void inicializar(String n, String a1, String a2, int e){
        nombre = n;
        apellido1 = a1;
        apellido2 = a2;
        edad = e;
    }

    public String nombreCompleto() {
        return nombre + " " + apellido1 + " " + apellido2;
    }

    public boolean esMayorEdad() {
        if (edad >= 18){
            return true;
        }else{
            return false;
        }
    }
}

```

2. Se escribe el código para utilizar la clase:

```

public class EjemploPoo{
    public static void main(String[] args) {

        //se crea un objeto de tipo Persona
        Persona p = new Persona();

        //se utilizan los métodos definidos en la clase Persona
        p.inicializar("Arsenio","Leante","Gomez",30);
        String nombre2 = p.nombreCompleto();
        if(p.esMayorEdad()){
            System.out.println(nombre2 + " es mayor de edad");
        }else{
            System.out.println(nombre2 + " no es mayor de edad");
        }
    }
}

```

En el ejemplo se ha creado un objeto llamado p de esta forma:

```
Persona p = new Persona();
```

En java cualquier variable u objeto que se crea debe tener un tipo. En este caso el tipo de p es *Persona*.

p es un objeto de tipo Persona

Una vez creado el objeto podemos invocar a sus métodos.

```
p.inicializar("Arsenio","Leante","Gomez",30);
```

Para llamar a un método se escribe **objeto.metodo()**, es decir, realizamos una operación sobre el objeto mediante un método.

En la actualidad existen una gran cantidad de lenguajes orientados a objetos. Algunos de estos lenguajes OO son:

C++, C#, VB.NET, Delphi, Java, PHP, Python, Ruby, Smalltalk, Golang...

## VENTAJAS DE LA PROGRAMACIÓN ORIENTADA A OBJETOS FRENTE A LA PROGRAMACIÓN PROCEDIMENTAL

La programación orientada a objetos se impuso como el estilo de programación dominante como evolución de la programación procedimental clásica.

Mediante la programación procedimental se pueden alcanzar excelentes resultados, pero cuando se aplica a problemas complejos, los resultados son menos satisfactorios. Algunos de los problemas que presenta el modelo tradicional de desarrollo de sistemas de información surgen:

- Cuando la complejidad del sistema o su ambigüedad no permite un análisis inicial completo.
- Si el sistema desarrollado no se adapta a las necesidades de los usuarios, por falta de entendimiento inicial entre usuarios y diseñadores.
- Requiere excesivos recursos dedicados a su mantenimiento.
- Resulta difícil llevar a cabo su modificación por cambios del entorno o de los requisitos iniciales.
- Cuando necesita incorporar el tratamiento de nuevos tipos de datos como imágenes, sonido, vídeo, etc.

La POO toma las mejores ideas incorporadas a la programación procedimental y las combina con nuevos y potentes conceptos que permiten organizar los programas en una forma más efectiva.

Algunas ventajas sobre la programación tradicional son:

- Los problemas se pueden descomponer en partes más pequeñas. Esto facilita la resolución de problemas complejos.
- Los programas son más comprensibles. Mediante la POO se obtiene un código más legible que refleja de una forma clara la relación entre cada concepto a desarrollar y cada objeto que interviene en dicho desarrollo.
- Reutilización de código. La herencia, es una de las claves de la POO para reutilizar y reducir código.
- Los programas son más resistentes al cambio. La modificación de un método no afectará a los otros ni a la estructura de datos.
- Los programas son más fáciles de mantener y modificar.

Si te ha sido útil compártelo

Post

No hay comentarios:

Publicar un comentario



Escribe tu comentario

[Entrada más reciente](#)

[Inicio](#)

[Entrada antigua](#)

Suscribirse a: [Enviar comentarios \(Atom\)](#)

#### LICENCIA



Programación Java by [Enrique García Hernández](#)

Esta obra está bajo una licencia [Creative Commons Reconocimiento-NoComercial-CompartirIgual 3.0 España License](#).

Para reconocer la autoría debes poner el enlace <http://puntocomnoesunlenguaje.blogspot.com.es>

Con la tecnología de [Blogger](#).

[Configuración de la privacidad y las cookies](#)

Gestionado por Google Cumple el TCF de IAB. ID de CMP: 300