

Java

POO: Métodos en Java





Métodos en Java

Cuando hablamos de programación Orientada a Objetos tenemos, implícitamente debemos vincular el concepto de Métodos...

Los métodos básicamente son fracciones de código dentro de nuestro sistema que permite realizar procedimientos o funciones específicas en nuestra aplicación.





Métodos en Java

Por medio de los métodos se empiezan a aplicar principios de modularidad, ya que manejamos de forma independiente procesos que pueden ser usados por el sistema...





Métodos en Java

ModificadorDeAlcance ModificadorDeUso TipoDeRetorno nombreMetodo (Parametros){
Cuerpo del metodo
} → (O de Acceso...)

- **ModificadoresDeAlcance:** Public, Private, Protected, Default
- **ModificadoresDeUso:** abstract, final, static
- **TipoDeRetorno :** int, double, String, void
- **nombreMetodo:** imprimirReporte(int id), obtenerResultado();
- **Cuerpo del Metodo :** Calculos, operaciones, sentencias





Métodos en Java

Java define una convención para la creación de métodos, esta convención se basa en el principio **camelCase** donde el nombre del método debe iniciar con la primera letra de la primera palabra en minúscula y cuando exista mas de una palabra debe iniciar con la primera letra en mayúscula....

Ej: registrarEdad();

Los métodos permiten realizar procesos y devolver o no valores, así como recibir o no ciertos datos para ser procesados



Java

Tipos de Métodos en Java





Tipos de Métodos en Java

En java podemos hablar básicamente de 4 tipos o estructuras de métodos en java:

- Métodos sin retorno y sin parámetros.
- Métodos sin retorno y con parámetros.
- Métodos con retorno y sin parámetros.
- Métodos con retorno y con parámetros.

Adicionalmente existe un método especial llamado método constructor.





Métodos sin retorno y sin parámetros.

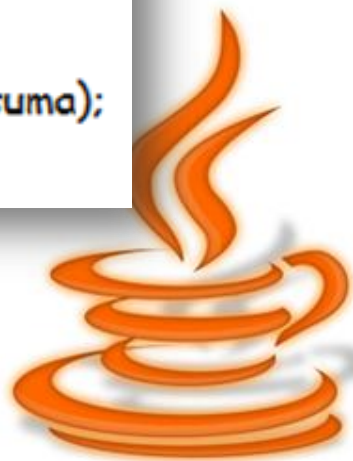
Estos tipos de métodos son aquellos que no devuelven ningún valor y no reciben ningún valor para ser procesado.

Diagram illustrating the components of a Java method signature:

- modificador de acceso** (access modifier): `public`
- tipo de retorno** (return type): `void`
- nombre del método** (method name): `sumar()`
- sin parámetros** (no parameters): Indicated by the empty parentheses in `sumar()`.

```
public void sumar(){  
    int x=5;  
    int y=10;  
    int suma=x+y;  
  
    System.out.println("El resultado de la suma es: "+suma);  
}
```

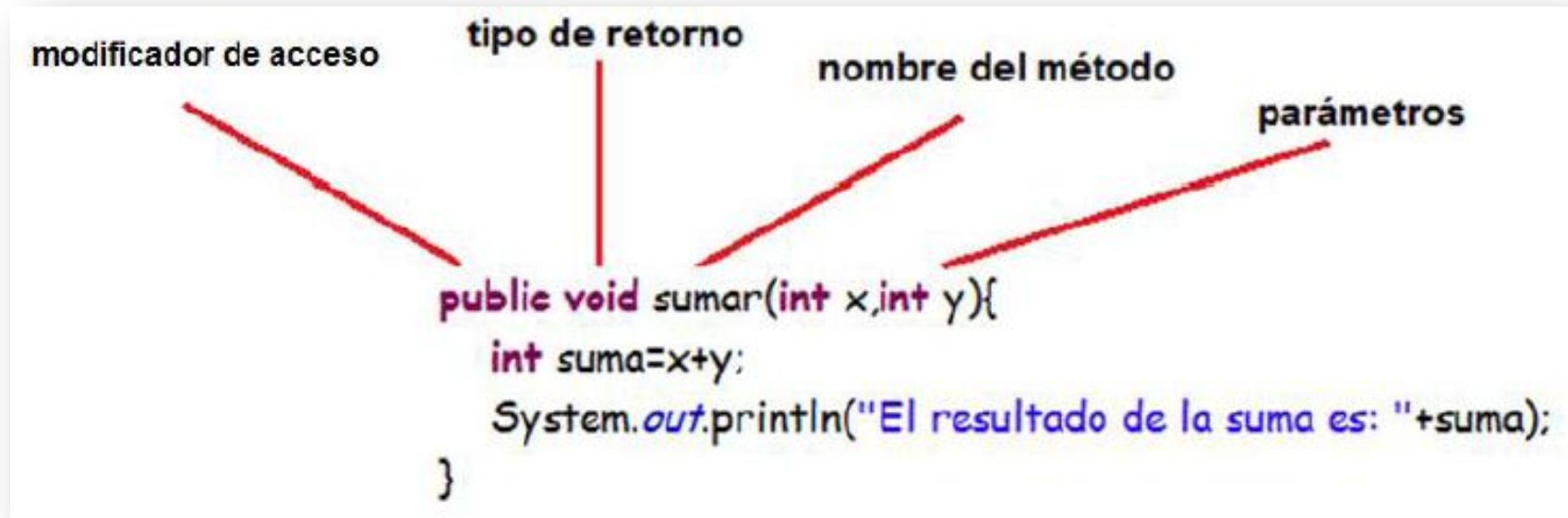
La palabra reservada `void` indica que el método no devuelve (retorna) ningún resultado.





Métodos sin retorno y con parámetros.

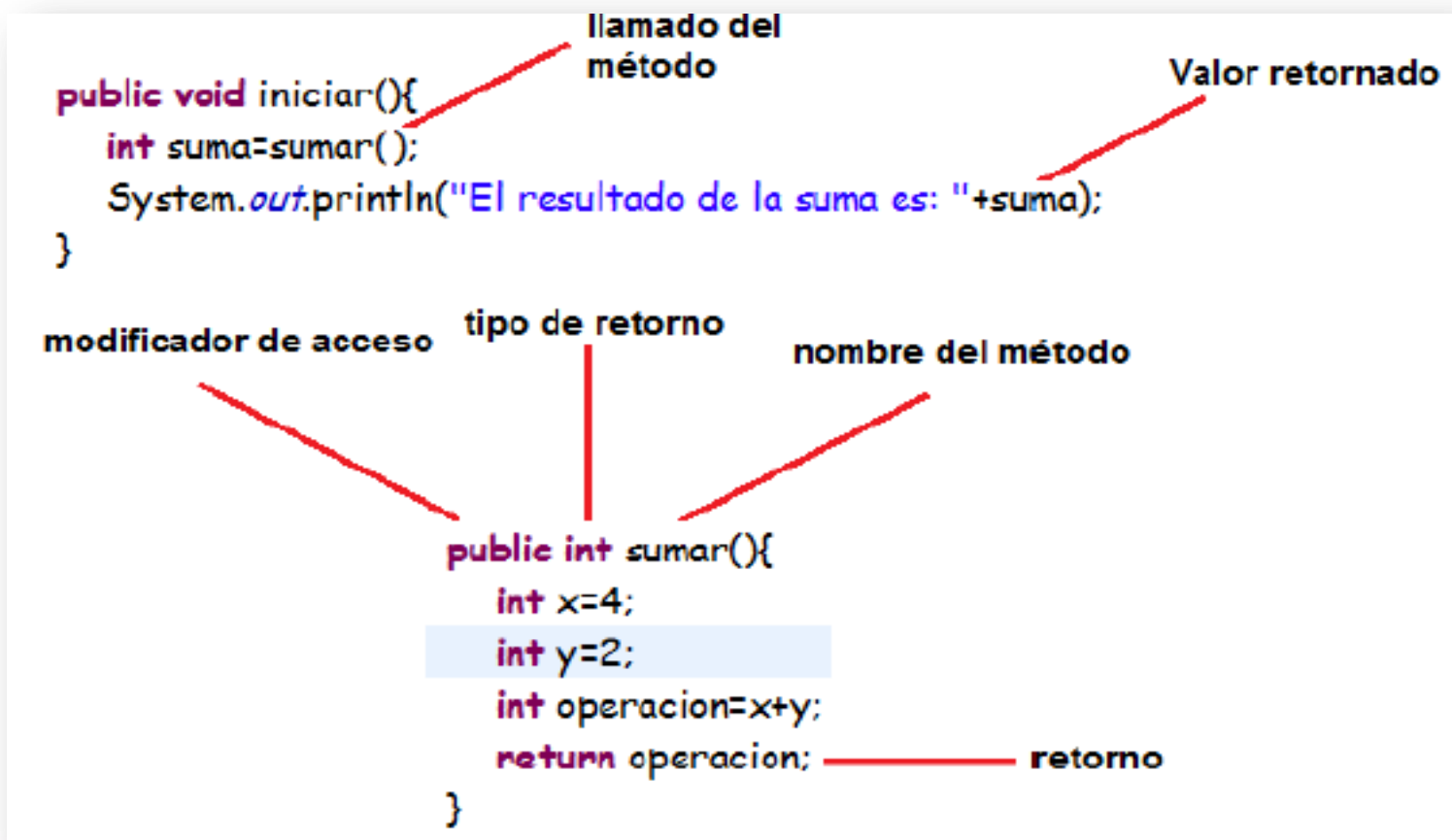
Estos tipos de métodos son aquellos que no devuelven ningún valor pero reciben uno o mas valores para ser procesados.





Métodos con retorno y sin parámetros.

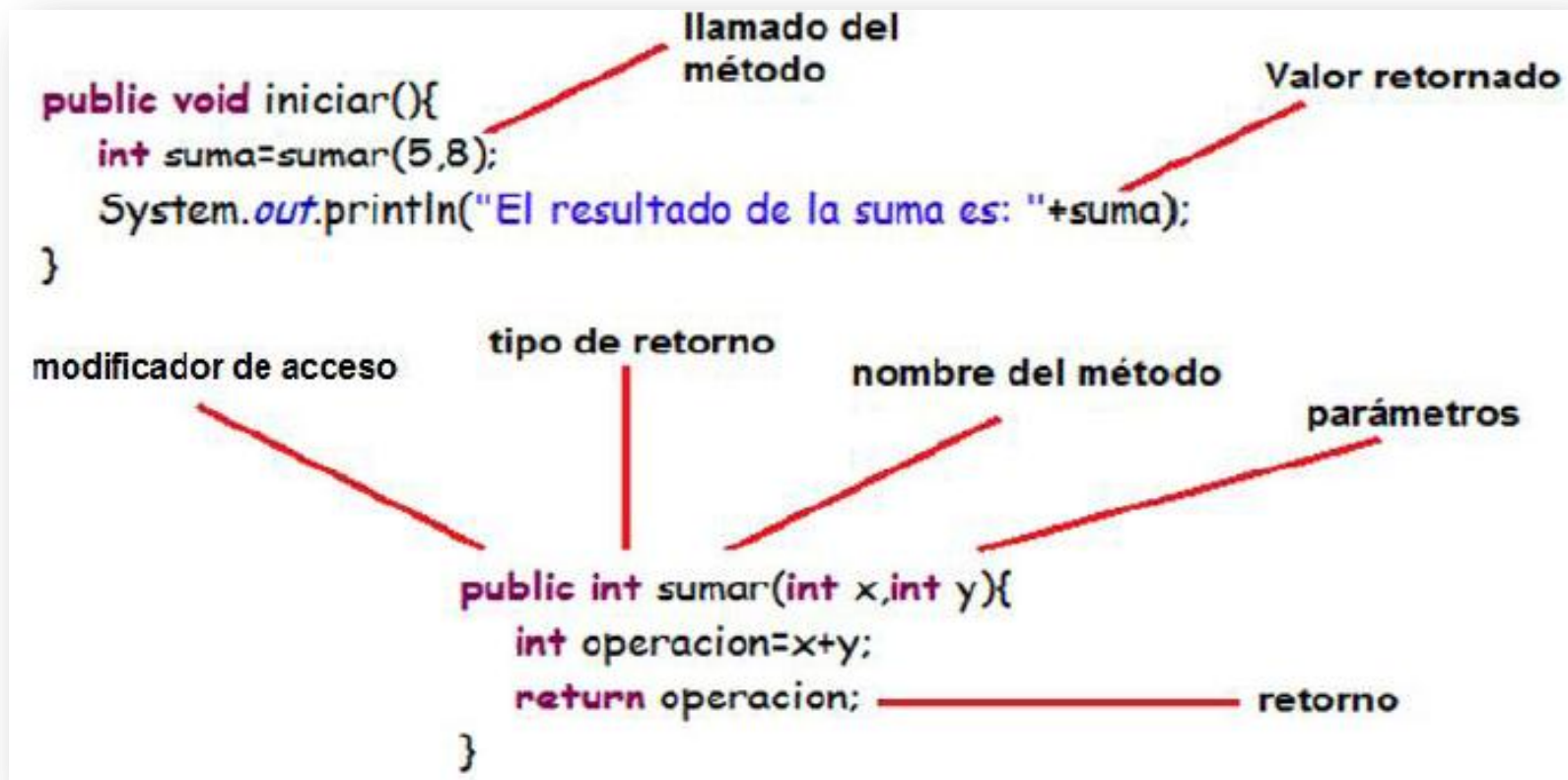
Estos tipos de métodos son aquellos que devuelven un valor pero no reciben ningún valor para ser procesado, este tipo de dato a devolver debe de ser del mismo tipo de dato del método.





Métodos con retorno y con parámetros.

Estos tipos de métodos son aquellos que devuelven un valor pero y reciben uno o mas valores para ser procesados, este tipo de dato a devolver debe de ser del mismo tipo de dato del método.



Java

Método Constructor





Método Constructor

El método constructor es un método especial que debe cumplir con las siguientes características:

- Debe llamarse exactamente igual a la clase
- Puede tener parámetros (opcional en caso de que se necesita)
- No tiene retorno, pero no necesita usar la palabra **void**.
- Permite la inicialización de componentes de la clase.
- Es el que permite la creación de objetos de una clase, al momento de decir new

Ej **new Animal()** -> **Animal()** es el constructor.





Método Constructor

Constructor



```
Animal miAnimal = new Animal();
```





Método Constructor

```
3 public class Clase {  
4  
5 public Clase(){  
6  
7 }  
8  
9 }  
10
```

El método constructor se llama igual a la clase...





Fuentes.

- Fundamentos de programación. Problemas resueltos de programación en lenguaje Java. José María Pérez Menor, Jesús Carretero Pérez, Félix García Carballeira, and José Manuel Pérez Lobato. Madrid: Paraninfo, 2003. p[1]-22
- <http://codejavu.blogspot.com/2013/05/conceptos-de-programacion-orientada.html>
- <http://codejavu.blogspot.com/2013/05/interfaces-en-java.html>
- JavaWorld – Revista Digital
- SCJP Sun Certified Programmer for Java 6

.

