

PROGRAMACIÓN II – JAVA

LETICIA CAROLINA DE LOS SANTOS BERMÚDEZ

lecasabe
CAPACITACIÓN PRÁCTICA

MÉTODOS

Si bien en Java se utiliza el término **método** para un conjunto de instrucciones definidas dentro de una clase, que realizan una determinada tarea y a las que podemos invocar mediante un nombre; es importante destacar que dentro de los métodos podemos encontrar diferencias y que se agrupan en **Funciones** y **Procedimientos**.

SINTAXIS

La sintaxis o estructura de un método se divide en cabecera y cuerpo.

SINTAXIS - CABECERA

Modificadores de acceso (`public`, `private`..) que determinan desde donde se puede utilizar el método. El modificador *static* indica que se trata de un método de clase, común para todos los objetos de la clase.

Tipo devuelto (cualquier tipo primitivo, no primitivo o `void`) que indica de que tipo es la salida del método, el resultado que se obtiene tras llamar al método desde el exterior. *void* se emplea cuando el método no devuelve ningún valor.

SINTAXIS - CABECERA

Nombre del método, que por convención (en Java) los nombres comienzan con minúscula.

Parámetros, si corresponde: entradas que necesita el método para realizar la tarea de la que es responsable. Cuando un método no tiene entradas, hay que poner paréntesis vacíos ().

SINTAXIS - CUERPO

Dentro del **cuerpo** se implementa el algoritmo necesario para realizar la tarea de la que el método es responsable.

Como cualquier bloque de código en Java, el cuerpo de un método ha de estar delimitada por llaves {}

FUNCIONES

Cuando asignamos un tipo de dato y un valor de retorno (*return*) estamos hablando de funciones. Las funciones pueden o no tener parámetro de entrada.

PROCEDIMIENTOS

Cuando el tipo de dato devuelto es *void* estamos hablando de procedimientos. Los procedimientos, al igual que las funciones, pueden o no tener parámetro de entrada.

PARÁMETROS

Es una variable utilizada para recibir valores de entrada en una rutina, subrutina o método. Dichos valores, que serán enviados desde la rutina invocante, son llamados argumentos.

EJEMPLO - FUNCIÓN SIN PARÁMETRO

```
3  
4 static int funcionSuma() {  
5  
6     int suma = 46+3;  
7     return suma;  
8  
9 }  
10
```

EJEMPLO - FUNCIÓN CON PARÁMETRO

```
11 static int funcionRestar(int n1, int n2) {  
12  
13     int resta = n1-n2;  
14     return resta;  
15 }  
16
```

INVOCACIÓN

Para poder ejecutar cualquier método es necesario invocarlos dentro del main.

```
16
17 public static void main(String[] args) {
18
19     funcionSuma();
20
21     int numero1 = 5;
22     int numero2 = 2;
23
24     funcionRestar(numero1, numero2);
25 }
```

EJEMPLO – PROCEDIMIENTO SIN PARÁMETRO

```
1 public class Procedimiento {  
2  
3     static void procedimientoSaludo() {  
4         System.out.println("Hola mundo!");  
5     }  
6     public static void main(String[] args) {  
7  
8         procedimientoSaludo();  
9  
10    }  
11  
12 }
```

UTILIDADES FUNDAMENTALES

- Organizar nuestro código para facilitar el mantenimiento.
- Ahorrar líneas de código.
- Permite repetir una acción varias veces pero sin repetir código sino simplemente invocándolo y, inclusive, pasar valores de entradas a parámetros diferentes.

FUENTE:

<https://www.programarya.com/Cursos/Java/Funciones>

<https://elvex.ugr.es/decsai/java/pdf/4B-methods.pdf>

<http://www.edu4java.com/es/java/metodos-en-java.html>