

Unidad 5. Modularidad

En Java se pueden utilizar **métodos** o **funciones**, las cuáles siempre van dentro de una clase. Aunque lo más común es que se declaren en clases independientes, también pueden declararse en la clase principal (la que tiene el método **main**).

Dentro de los tipos de métodos o funciones en Java están aquellas que son de tipo **void** (que no devuelven nada) o aquellas que tienen un tipo de dato (int, String, boolean, etc.) y que **devuelven un valor**, para lo cual se debe utilizar la cláusula **return**.

Tanto los métodos void como las funciones que devuelven algún valor, pueden recibir parámetros o argumentos.

Métodos de tipo void.

Los métodos tipo **void** son aquellos que no devuelven ningún tipo de valor. Solo ejecutan código que tengan en su interior cuando son llamados. Este tipo de métodos pueden recibir o no parámetros o argumentos, según se necesite.

Para entender los métodos tipo void se realizará el siguiente ejemplo.

Ejemplo 5.1

Realiza el programa en JAVA tal que, dado como datos la base y la altura de un rectángulo, calcule e imprima el perímetro y la superficie del mismo.

Perímetro = $2 * (\text{base} + \text{altura})$

Superficie = $\text{base} * \text{altura}$

Datos: base, altura

Donde: **base** y **altura** son variables de tipo entero que representan las dimensiones de un rectángulo.
perímetro y **superficie** son variables de tipo entero que representan los cálculos por realizar.

1. Crear el proyecto Ejemplo 5.1 y en el método principal (**main**) solo escribir un par de mensajes.

```
1  ...5 lines
6
7  /**...4 lines */
11 public class Ejemplo51 {
12
13
14
15
16     public static void main(String[] args) {
17         // TODO code application logic here
18         System.out.println("Perímetro y superficie de un rectángulo");
19
20         System.out.println("Fin del programa");
21     }
```

2. Crear el método **calculos** dentro de la clase **Ejemplo51**, pero antes del método **main**.

```
1  + ...5 lines
6
7  + /**...4 lines */
11 public class Ejemplo51 {
12
13     public static void calculos() {
14
15     }
16
17     public static void main(String[] args) {
18         // TODO code application logic here
19         System.out.println("Perímetro y superficie de un rectángulo");
20
21         System.out.println("Fin del programa");
22     }
23
24 }
```

Método *calculos*

Llamar o invocar a los métodos.

Es muy sencillo invocar o llamar los métodos en Java, sólo se necesita el nombre del método o función, y enviarle el valor de los parámetros en caso de tenerlos.

3. Hacer la llamada del método **calculos** en el método principal (**main**).

```
1  + ...5 lines
2  import java.util.*;
7  + /**...4 lines */
11 public class Ejemplo51 {
12
13     public static void calculos() {
14
15     }
16
17     public static void main(String[] args) {
18         // TODO code application logic here
19         System.out.println("Perímetro y superficie de un rectángulo");
20
21         calculos();
22
23         System.out.println("Fin del programa");
24     }
25
26 }
```

Método *calculos*

llamada al método *calculos*

Hasta este punto, el programa aun no hace nada, por lo que es necesario codificar el método **calculos**, en el cual se declararán las variables, se leerán la base y la altura, se harán los cálculos y se mostrarán los resultados.

4. Codificar las sentencias del método **calculos**.

```
1  ...5 lines
6  import java.util.*;
7  /**...4 lines */
11 public class Ejemplo51 {
12
13     public static void calculos(){
14         int base, altura, perimetro, superficie;
15         Scanner Leer = new Scanner(System.in);
16         System.out.print("Introduce la base: ");
17         base = Leer.nextInt();
18         System.out.print("Introduce la altura: ");
19         altura = Leer.nextInt();
20
21         perimetro = 2*(base+altura);
22         superficie = base * altura;
23
24         System.out.println("El perímetro es: " + perimetro);
25         System.out.println("La superficie es: "+ superficie);
26     }
27
28     public static void main(String[] args) {
29         // TODO code application logic here
30         System.out.println("Perímetro y superficie de un rectángulo");
31
32         calculos();
33
34         System.out.println("Fin del programa");
35     }
36
37 }
```

5. Finalmente ejecutar el programa. Si se introduce 10 para la base y 5 para la altura el resultado sería:

```
Perímetro y superficie de un rectángulo
Introduce la base: 10
Introduce la altura: 5
El perímetro es: 30
La superficie es: 50
Fin del programa
BUILD SUCCESSFUL (total time: 6 seconds)
```

Método calculos

Método main