

## Unidad 5. Modularidad

### Métodos que retornan valores (funciones)

En algunos lenguajes de programación, distintos a Java, existen unos bloques de código llamados funciones. En Java, esas funciones son los *métodos que retornan valores*. Estos métodos normalmente pueden recibir parámetros o argumentos (aunque rara vez algunas no reciben nada) y su principal característica es que **siempre** regresan un valor mediante la instrucción **return**.

Los métodos que retornan valores (funciones) no utilizan el modificador **void**, en su lugar se especifica el **tipo de dato** que devolverá el método (función). Es importante saber que, el tipo de dato devuelto por el método debe ser almacenado en una variable con el mismo tipo de dato del método.

#### Ejemplo 5.3

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 5.3 y en el método principal (**main**) se debe declarar las variables, que permitan leer la base y la altura del rectángulo, así como las variables que recibirán los valores de los métodos.

```
1  + ...5 lines
6  - import java.util.*;
7  + /**...4 lines */
11 public class Ejemplo53 {
12
13  -     public static void main(String[] args) {
14      // TODO code application logic here
15      int base, altura, longitud, area;
16      String texto;
17      Scanner Leer = new Scanner(System.in);
18      System.out.println("Perímetro y superficie de un rectángulo");
19      System.out.print("Introduce la base: ");
20      base = Leer.nextInt();
21      System.out.print("Introduce la altura: ");
22      altura = Leer.nextInt();
23
24  }
25
26 }
```

2. Crear el método **perimetro** de tipo **int**, dentro de la clase **Ejemplo53**, pero antes del método **main**.

El método debe especificar los parámetros que va a recibir (*base* y *altura*) los cuales utilizará para calcular el perímetro y entonces devolver el resultado.

### Consideraciones sobre los métodos que devuelven valores

- El tipo de dato del método debe ser igual al tipo de dato que devolverá dicho método.
- Todos los métodos que devuelven valores deben utilizar la instrucción **return** con el valor a devolver.
- Los métodos que devuelven valores **NO** utilizan el modificador **void**.

```
1  + ...5 lines
6  - import java.util.*;
7  + /**...4 lines */
11 public class Ejemplo53 {
12
13     public static int perimetro(int b, int a) {
14         int per;
15         per = 2 * (b + a);
16         return per;
17     }
18
19     public static void main(String[] args) {
20         // TODO code application logic here
21         int base, altura, longitud, area;
22         String texto;
23         Scanner Leer = new Scanner(System.in);
24         System.out.println("Perímetro y superficie de un rectángulo");
25         System.out.print("Introduce la base: ");
26         base = Leer.nextInt();
27         System.out.print("Introduce la altura: ");
28         altura = Leer.nextInt();
29
30     }
31
32 }
```

Diagram annotations:

- Green arrow pointing to `int` in `perimetro`: tipo de dato que devuelve el método
- Green arrow pointing to `int b, int a`: parámetros que recibe el método
- Green arrow pointing to `return per`: El tipo de dato devuelto debe ser igual al del método

## Llamar o invocar a los métodos que regresan valores.

Normalmente cuando se invoca o llama a un método que regresa un valor, se utiliza una variable que almacena el valor devuelto por el método. La variable debe ser del mismo tipo de dato que el método.

3. Hacer la llamada del método **perimetro** en el método principal (**main**) y mostrar el resultado.

```
1  + ...5 lines
6  - import java.util.*;
7  + /**...4 lines */
11 public class Ejemplo53 {
12
13  -     public static int perimetro(int b, int a){
14      -         int per;
15      -         per = 2 * (b + a);
16      -         return per;
17      -     }
18
19  -     public static void main(String[] args) {
20      -         // TODO code application logic here
21      -         int base, altura, longitud, area;
22      -         String texto;
23      -         Scanner Leer = new Scanner(System.in);
24      -         System.out.println("Perímetro y superficie de un rectángulo");
25      -         System.out.print("Introduce la base: ");
26      -         base = Leer.nextInt();
27      -         System.out.print("Introduce la altura: ");
28      -         altura = Leer.nextInt();
29      -         longitud = perimetro(base, altura);
30      -         System.out.println("El perímetro es: "+ longitud);
31      -     }
32
33
34
35
36 }
```

- Llamada al método perimetro  
- El método regresa el valor y lo almacena en la variable longitud





4. Crear el método **superficie** de tipo **int**. El método debe especificar los parámetros que va a recibir (*base* y *altura*) los cuales utilizará para calcular el área del rectángulo.
5. Crear el método **mensaje** de tipo **String** para que regrese una cadena de texto especificando el fin del programa.

```
1  + ...5 lines
6  - import java.util.*;
7  + /**...4 lines */
11 public class Ejemplo53 {
12
13  - public static int perimetro(int b, int a){
14      int per;
15      per = 2 * (b + a);
16      return per;
17  }
18
19  - public static int superficie(int b, int a){
20      int sup;
21      sup = b * a;
22      return sup;
23  }
24
25  - public static String mensaje(){
26      return "Fin del programa";
27  }
28
```

6. Hacer la llamada de los métodos **superficie** y **mensaje** en el método principal (**main**) y mostrar los resultados

```
1  + ...5 lines
6  - import java.util.*;
7  + /**...4 lines */
11 public class Ejemplo53 {
12
13  -     public static int perimetro(int b, int a){
14      |         int per;
15      |         per = 2 * (b + a);
16      |         return per;
17      |     }
18
19  -     public static int superficie(int b, int a){
20      |         int sup;
21      |         sup = b * a;
22      |         return sup;
23      |     }
24
25  -     public static String mensaje(){
26      |         return "Fin del programa";
27      |     }
28
29  -     public static void main(String[] args) {
30      |         // TODO code application logic here
31      |         int base, altura, longitud, area;
32      |         String texto;
33      |         Scanner Leer = new Scanner(System.in);
34      |         System.out.println("Perímetro y superficie de un rectángulo");
35      |         System.out.print("Introduce la base: ");
36      |         base = Leer.nextInt();
37      |         System.out.print("Introduce la altura: ");
38      |         altura = Leer.nextInt();
39      |
40      |         longitud = perimetro(base, altura);
41      |         System.out.println("El perímetro es: "+ longitud);
42      |
43      |         area = superficie(base, altura);
44      |         System.out.println("La superficie es: "+ area);
45      |
46      |         texto = mensaje();
47      |         System.out.println(texto);
48      |     }
49  }
50
51 }
```

7. Finalmente, ejecutar el programa e identificar lo que sucede en cada uno de los métodos (main, perimetro, superficie y mensaje) si se introduce 10 para la base y 5 para la altura.



```
run:
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: 9 seconds)
```