Unidad 5. Modularidad

Métodos de tipo void con paso de parámetros.

Los métodos tipo **void** también pueden recibir valores (parámetros), los cuáles se utilizan para realizar operaciones en el interior del método. En Java, el paso de parámetros se realiza siempre por valor.

En este ejemplo, se verá cómo se crea el método en java con paso de parámetros para calcular el área y perímetro de un rectángulo.

Ejemplo 5.2
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 * (base + altura)
Superficie = base * altura

Datos: base, altura

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

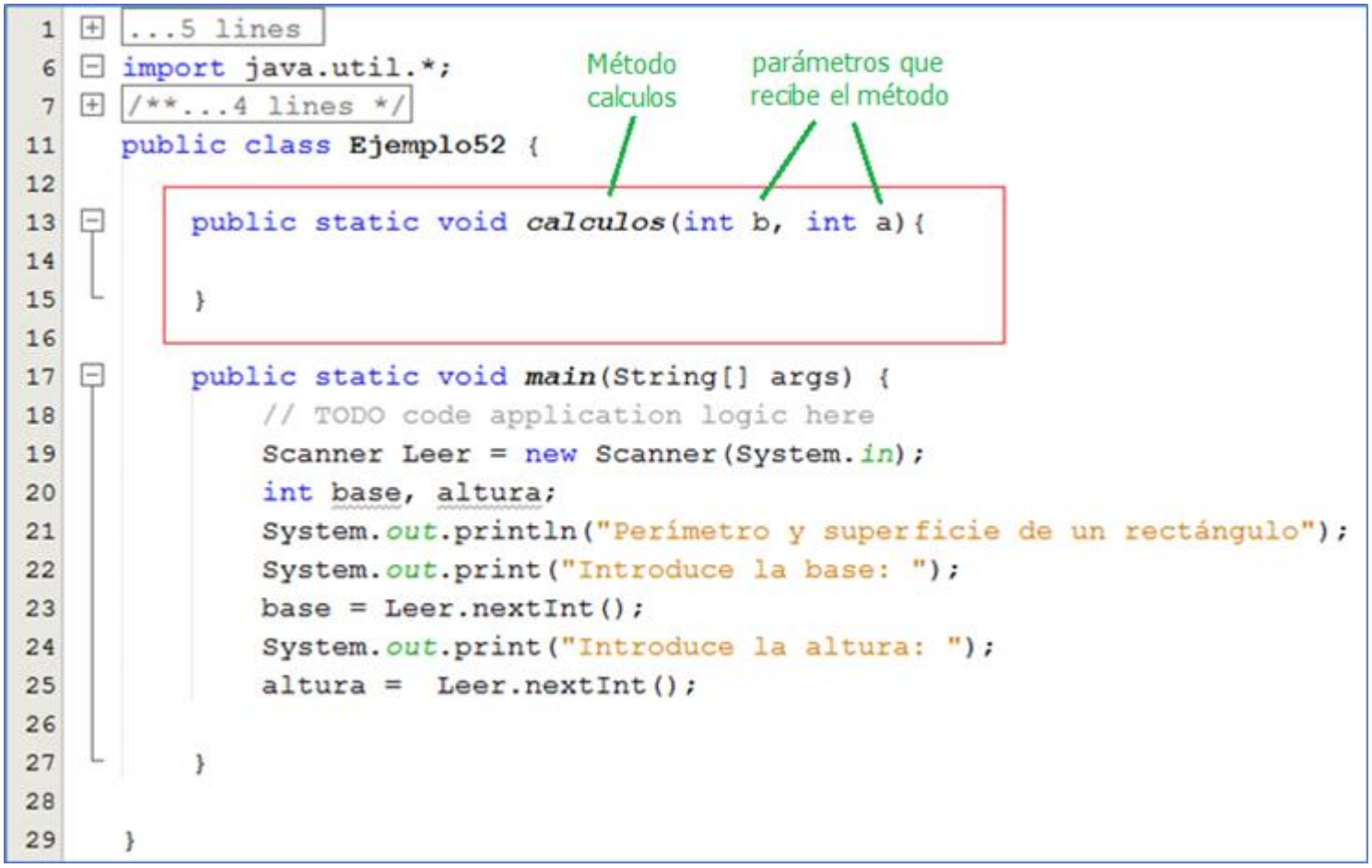
1. Crear el proyecto Ejemplo 5.2 y en el método principal (**main**) solo se debe declarar las variables, que permitan leer la base y la altura del rectángulo.

```
1  ...5 lines
6  import java.util.*;
7  /**...4 lines */
11 public class Ejemplo52 {
12
13     public static void main(String[] args) {
14         // TODO code application logic here
15         Scanner Leer = new Scanner(System.in);
16         int base, altura;
17         System.out.println("Perímetro y superficie de un rectángulo");
18         System.out.print("Introduce la base: ");
19         base = Leer.nextInt();
20         System.out.print("Introduce la altura: ");
21         altura =  Leer.nextInt();
22
23     }
24
25 }
```

2. Crear el método **calculos** con los parámetros para recibir la *base* y la *altura*, dentro de la clase **Ejemplo52**, pero antes del método **main**.

Consideraciones sobre los *parámetros o argumentos*

- Un método o una función, puede tener cualquier cantidad de parámetros.
- Los parámetros de una función deben tener un tipo y un nombre que los identifica. El tipo del parámetro puede ser cualquiera y no tiene relación con el tipo del método.
- Si un método tiene más de un parámetro, cada uno de ellos debe ir separado por una coma.
- Los parámetros que se reciben pueden ser por valor o por referencia, esto implica que, si se modifican los valores recibidos al interior del método, estos pueden mantener sus cambios o no después de ejecutado el método.



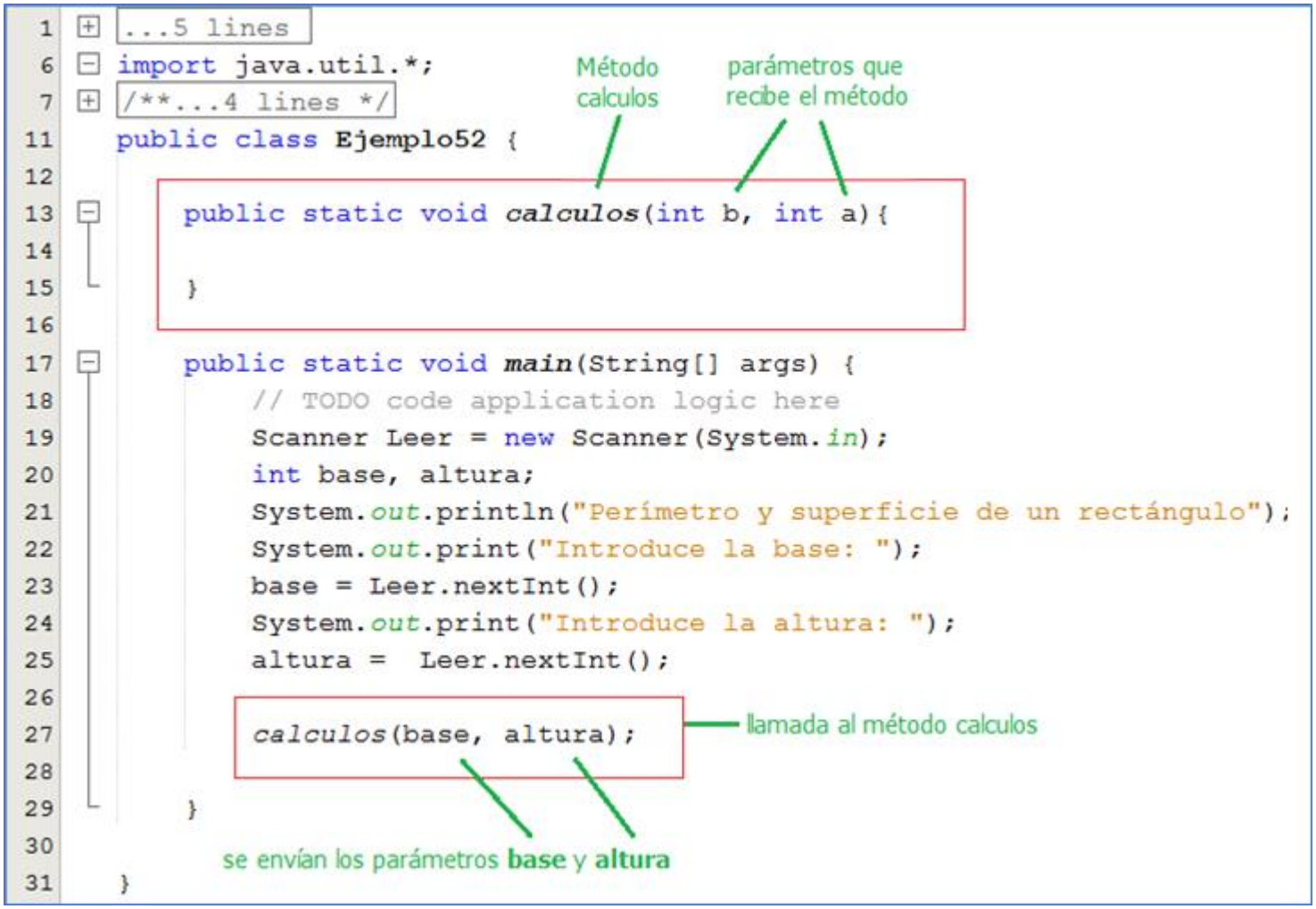

```
1  ...5 lines
6  import java.util.*;
7  /**...4 lines */
11 public class Ejemplo52 {
12
13     public static void calculos(int b, int a){
14
15     }
16
17     public static void main(String[] args) {
18         // TODO code application logic here
19         Scanner Leer = new Scanner(System.in);
20         int base, altura;
21         System.out.println("Perímetro y superficie de un rectángulo");
22         System.out.print("Introduce la base: ");
23         base = Leer.nextInt();
24         System.out.print("Introduce la altura: ");
25         altura =  Leer.nextInt();
26
27     }
28
29 }
```

Llamar o invocar a los métodos con parámetros.

Para invocar o llamar los métodos que tienen parámetros, sólo se necesita especificar el nombre del método o función y enviarle los parámetros.

- La cantidad, tipo y orden de los parámetros que se envían debe coincidir con la cantidad, tipo y orden de los parámetros utilizados en el método.

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



Hasta este punto, el programa solo lee la base y la altura en el método **main**. Lo que hace falta es realizar los cálculos y mostrar los resultados, lo cual se codificará en el método **calculos**.

4. Codificar las sentencias del método **calculos**. para calcular y mostrar el perímetro y superficie del rectángulo.

```
1  ...5 lines
6  import java.util.*;
7  /**...4 lines */
11 public class Ejemplo52 {
12
13     public static void calculos(int b, int a){
14         int perimetro, superficie;
15         perimetro = 2 * (b + a);
16         superficie = b * a;
17
18         System.out.println("El perímetro es: " + perimetro);
19         System.out.println("La superficie es: "+  superficie);
20     }
21
22     public static void main(String[] args) {
23         // TODO code application logic here
24         Scanner Leer = new Scanner(System.in);
25         int base, altura;
26         System.out.println("Perímetro y superficie de un rectángulo");
27         System.out.print("Introduce la base: ");
28         base = Leer.nextInt();
29         System.out.print("Introduce la altura: ");
30         altura =  Leer.nextInt();
31
32         calculos(base, altura);
33
34     }
35
36 }
```

5. Finalmente, ejecutar el programa e identificar lo que sucede en el método main y en el método calculos si se introduce 8 para la base y 4 para la altura.

```
run:
Perímetro y superficie de un rectángulo
Introduce la base: 8
Introduce la altura: 4
El perímetro es: 24
La superficie es: 32
BUILD SUCCESSFUL (total time: 10 seconds)
```

método main

método calculos