

## 14. Tipos de métodos

### Objetivos:

- Profundizar en el concepto de método de una clase e indicar los tipos de métodos en Java: métodos de instancia y métodos de clase
- Interpretar el código fuente de una aplicación Java donde aparecen métodos de distintos tipos
- Construir una aplicación Java sencilla, convenientemente especificada, que emplee clases con diferentes métodos.

Un método es una abstracción de una operación que puede hacer o realizarse con un objeto. Una clase puede declarar cualquier número de métodos que lleven a cabo operaciones de lo más variado con los objetos. En esta sección los métodos se clasifican en dos grupos: los métodos *de instancia* y los métodos *de clase*. Además se cierra el capítulo con los métodos de clase o *estáticos* de la clase `Math`.

### 14.1. Métodos de instancia

Las clases pueden incluir en su declaración muchos métodos o no declarar ninguno. Los métodos pueden clasificarse en métodos *de instancia* y métodos *de clase*.

Los *métodos de instancia* operan sobre las variables de instancia de los objetos pero también tienen acceso a las variables de clase. La sintaxis de llamada a un método de instancia es:

```
idReferencia.idMetodo(parametros); // Llamada típica a un método de instancia
```

Todas las instancias de una clase comparten la misma implementación para un método de instancia. La instancia que hace la llamada al método es siempre un parámetro o argumento implícito. Dentro de un método de instancia, el identificador de una variable de instancia hace referencia al atributo de la instancia concreta que hace la llamada al método (suponiendo que el identificador del atributo no ha sido *ocultado* por el de un parámetro).

En el ejemplo anterior en la declaración de la clase `CuentaBancaria`, los métodos `saldo` y `transferencia` son métodos de instancia.

```
public double saldo() {
    return saldo;
}
public void transferencia( CuentaBancaria origen ) {
    saldo += origen.saldo;
    origen.saldo=0;
}
```

Ejemplos de llamada a estos métodos dentro de `PruebaCuentaBancaria`:

```
CuentaBancaria c1 = new CuentaBancaria();
CuentaBancaria c2 = new CuentaBancaria(20.0);
c1.transferencia(c2);
System.out.println("Cuenta con: " + c1.saldo() + " euros");
```

## 14.2. Métodos de clase

En principio, los *métodos de clase* no operan sobre las variables de instancia de los objetos. Los *métodos de clase* pueden trabajar con las **variables de clase** pero no pueden acceder a las variables de instancia declaradas dentro de la clase, a no ser que se crea una nueva instancia y se acceda a las variables de instancia a través del nuevo objeto. Los métodos de clase también pueden ser llamados precediéndolos con el identificador de la clase, sin necesidad de utilizar el de una instancia.

```
IdClase.idMetodo(parametros); // Llamada típica a un método de clase
```

La palabra `static` determina la declaración de un método de clase. Por defecto, si no se indica la palabra `static`, el método declarado se considera un método de instancia.

En el ejemplo anterior en la declaración de la clase `CuentaBancaria`, el método `incCuentas` es un método de clase.

```
public static void incCuentas () {
    totalCuentas++;
}
```

Un ejemplo de llamada a este método dentro de `PruebaCuentaBancaria` sería:

```
CuentaBancaria.incCuentas();
```

Las diferencias entre los métodos de instancia y los de clase se resumen en la Tabla 14.1.

Tabla 14.1. Diferencias entre los métodos de instancia y los métodos de clase

Métodos...	Modificador en la declaración	Sintaxis de llamada	Operan normalmente...
... de instancia	-- (por defecto)	<i>instancia.metodo(parametros)</i>	Sobre variables de instancia
... de clase	<code>static</code>	<i>Clase.metodo(parametros)</i>	Sobre variables de clase o sobre otros datos

Los métodos de clase o *estáticos* se pueden considerar equivalentes a las rutinas (globales) de otros lenguajes de programación como Pascal o C. Como ejemplos típicos de métodos estáticos pueden indicarse los métodos de Java correspondientes a las funciones matemáticas `sin`, `cos`, `exp`, `pow...` de la clase `java.lang.Math` (Tabla 14.2). Las llamadas a estos métodos se realizan anteponiendo el identificador de la clase `Math` al identificador del método: `Math.sin(angulo)...`

Tabla 14.2. Métodos estáticos de la clase Math

Modificadores	Identificador y parámetros	Función
static double	<b>abs</b> (double a)	Devuelve el valor absoluto de un dato de tipo double.
static float	<b>abs</b> (float a)	Devuelve el valor absoluto de un dato de tipo float.
static int	<b>abs</b> (int a)	Devuelve el valor absoluto de un dato de tipo int.
static long	<b>abs</b> (long a)	Devuelve el valor absoluto de un dato de tipo long.
static double	<b>acos</b> (double a)	Devuelve el valor del arco coseno de un ángulo entre 0 y $\pi$ .
static double	<b>asin</b> (double a)	Devuelve el valor del arco seno de un ángulo entre $-\pi/2$ y $\pi/2$ .
static double	<b>atan</b> (double a)	Devuelve el arco tangente de un ángulo entre $-\pi/2$ y $\pi/2$ .
static double	<b>atan2</b> (double a, double b)	Devuelve el ángulo en polares correspondientes a las coordenadas rectangulares.
static double	<b>ceil</b> (double a)	Devuelve el menor entero (en formato double) que no es menor que el parámetro.
static double	<b>cos</b> (double a)	Devuelve el coseno de un ángulo
static double	<b>exp</b> (double a)	Devuelve el valor de la función exponencial ( $e^a$ )
static double	<b>floor</b> (double a)	Devuelve el mayor entero (en formato double) que no es mayor que el parámetro.
static double	<b>IEEEremainder</b> (double f1, double f2)	Devuelve el resto (en formato double) de la división entre los dos parámetros según la especificación IEEE 754
static double	<b>log</b> (double a)	Devuelve el logaritmo neperiano de un valor.
static double	<b>max</b> (double a, double b)	Devuelve el mayor de dos valores de tipo double.
static float	<b>max</b> (float a, float b)	Devuelve el mayor de dos valores de tipo float.
static int	<b>max</b> (int a, int b)	Devuelve el mayor de dos valores de tipo int.
static long	<b>max</b> (long a, long b)	Devuelve el mayor de dos valores de tipo long.
static double	<b>min</b> (double a, double b)	Devuelve el menor de dos valores de tipo double.
static float	<b>min</b> (float a, float b)	Devuelve el menor de dos valores de tipo float.
static int	<b>min</b> (int a, int b)	Devuelve el menor de dos valores de tipo int.
static long	<b>min</b> (long a, long b)	Devuelve el menor de dos valores de tipo long.
static double	<b>pow</b> (double a, double b)	Devuelve el valor del primer parámetro elevado al segundo
static double	<b>random</b> ()	Devuelve un valor aleatorio de tipo double mayor o igual que cero y menor que 1.
static double	<b>rint</b> (double a)	Devuelve el valor entero (en formato double) más cercano.
static long	<b>round</b> (double a)	Devuelve el valor entero (long) más cercano al parámetro.
static int	<b>round</b> (float a)	Devuelve el valor entero (int) más cercano al parámetro.
static double	<b>sin</b> (double a)	Devuelve el seno de un ángulo
static double	<b>sqrt</b> (double a)	Devuelve la raíz cuadrada positiva del parámetro double.
static double	<b>tan</b> (double a)	Devuelve la tangente de un ángulo
static double	<b>toDegrees</b> (double angulo)	Devuelve el resultado de la conversión de un ángulo dado en radianes a grados
static double	<b>toRadians</b> (double angulo)	Devuelve la conversión de un ángulo dado en grados a radianes