

Tema 3

FUNCIONES

Autores: Carlos Cacho y Raquel Torres
(adaptación Antonio Hernández)

1º Desarrollo de Aplicaciones Multiplataforma
Módulo formativo - Programación

Introducción

La mejor forma de crear y mantener un programa grande es construirlo a partir de piezas más pequeñas o módulos. Cada uno de los cuales es más manejable que el programa en su totalidad.

Introducción

Las **funciones** (subprogramas) son utilizadas para **evitar la repetición de código** en un programa al poder ejecutarlo desde varios puntos de un programa con sólo invocarlo.

Utilidad principal de las funciones:

- Agrupar código que forma una entidad propia o una idea concreta.
- Agrupar código que se necesitará varias veces en un programa, con la misión de no repetir código.
- Dividir el código de un programa grande en subprogramas (funciones), cada uno de ellos especializados en resolver una parte del problema.

Características de las funciones:

- Se definen mediante un nombre único que representa el bloque de código.
- Pueden ser llamadas (ejecutadas) desde cualquier parte del código.
- Se les puede pasar valores para que los procesen de alguna forma.
- Pueden devolver un resultado para ser usado desde donde se les llamó.

Declaración de una función

- Declarar una función simplemente significa crearla para que luego pueda ser llamada (utilizada) desde otro lugar del código de nuestro programa.
- Una función se estructura en **cabecera** y **cuerpo**.

Cabecera de una función

Se declara en una sola línea y se compone de:

- **Modificadores de función:** Existen muchos pero los veremos en futuras unidades. Por ahora solo utilizaremos public static).
- **Tipo devuelto:** El tipo de dato que devolverá la función, como por ejemplo int, double, char, boolean, String, etc. Si la función no devuelve nada se indica mediante void.

Cabecera de una función

- **Nombre de la función:** Identificador único para llamar a la función.
- **Lista de parámetros:** Indica los tipos y nombres de los datos que se le pasarán a la función cuando sea llamada. Pueden ser varios o ninguno.

[Modif_de_función] Tipo_devuelto Nombre_de_función (lista_de_parámetros)

{

...

}

Cuerpo de una función

El **cuerpo** es un bloque de código entre llaves { ... } que se ejecutará cuando desde otra parte del código utilicemos la función.

- **Ejemplo de funciones:**

```
public static void imprimeHolaMundo() {  
    System.out.println("Hola mundo");  
}
```

Esta función se llama 'imprimeHolaNombre',
tiene como parámetro de entrada un dato **String** llamado
'nombre' y no devuelve **nada**.

Cuando la llamemos imprimirá por pantalla el texto "Hola "
seguido del String nombre que se le pase como parámetro.

```
public static void imprimeHolaNombre(String nombre) {  
    System.out.println("Hola " + nombre);  
}
```

Esta función se llama 'doble',
tiene como parámetro de entrada un dato **int** llamado 'a' y
devuelve un dato de tipo **int**.

Cuando la llamemos calculará el doble de 'a' y lo devolverá
(con el return).

```
public static int doble(int a) {  
    int resultado = a * 2;  
    return resultado;  
}
```

Esta función se llama 'maximo',

tiene dos parámetros de entrada de tipo **double** llamados 'valor1' y 'valor2' y devuelve un dato de tipo **double**. Cuando la llamemos calculará el máximo entre 'valor1' y 'valor2' y lo devolverá .

```
public static double maximo(double valor1, double valor2) {  
    double max;  
    if (valor1 > valor2) {  
        max = valor1;  
    }  
    else {  
        max = valor2;  
    }  
    return max;  
}
```

Esta función se llama 'esPar',
tiene un parámetros de entrada de tipo **int** llamado 'numero' y devuelve un dato de tipo **boolean**.

```
public static boolean esPar(int numero) {  
    if (numero % 2 == 0) {  
        return true;  
    } else {  
        return false;  
    }  
}
```

Declaración de una función

- Es importante saber que **las funciones se declaran dentro de 'class' pero fuera del 'main' (una función no se declara dentro de otra).**
- Sí, el 'main' donde siempre has programando hasta ahora es una función, pero un poco especial: **'main' es la función principal, el punto de inicio de un programa.**

Declaración de una función

- Es obligatorio que todo programa Java tenga una función main.
- Si te fijas, es una función que recibe como parámetro un `String[]` y no devuelve nada (`void`).
- Las funciones que hemos declarado por sí solas no hacen nada, simplemente están ahí esperando a que sean **llamadas**

Llamada a una función

- Las funciones pueden ser invocadas o llamadas desde cualquier otra función.
- Cuando se invoca una función el flujo de ejecución salta a la función (pasándole los parámetros si los hubiera), se ejecutan las instrucciones de la función y por último vuelve al punto que llamó a la función para seguir ejecutándose.

Llamada a una función

- Las funciones se invocan con su nombre, pasando la lista de parámetros entre paréntesis. Si no tiene parámetros han de ponerse los paréntesis igualmente.
- Si la función devuelve un valor, para recogerlo hay que asignarlo a una variable o utilizarlo de algún modo (pueden combinarse funciones en expresiones e incluso pasarlo a otras funciones).

```
public static void main(String[] args) {
```

```
    // No tiene parámetros ni devuelve valor. Simplemente imprime "Hola Mundo"
```

```
    imprimeHolaMundo();
```

```
    // Es habitual llamar a una función y guardar el valor devuelto en una variable
```

```
    int a = doble(10); // a = 20 (10*2)
```

```
    int b = multiplica(3, 5); // b = 15 (3*5)
```

```
    // Pueden pasarse variables como parámetros
```

```
    int c = doble(a); // c = 40 (20*2)
```

```
    int d = multiplica(a, b); // d = 300 (20*15)
```

```
    // Pueden combinarse funciones y expresiones
```

```
    int e = doble(4) + multiplica(2,10); // e = 8 + 20
```

```
    System.out.println("El doble de 35 es " + doble(35) ); // "El doble de 35 es 70"
```

```
    System.out.println("12 por 12 es " + multiplica(12,12) ); // "12 por 12 es 144"
```

```
}
```

Paso por valor

Parámetros de tipo simple (int, double, boolean, char, etc.) **se pasan por valor.**

El valor se copia al parámetro y por lo tanto si se modifica dentro de la función esto no afectará al valor fuera de ella porque **son variables distintas.**

Paso por valor

```
public static void main(String[] args) {  
    int a = 10;  
    System.out.println("Valor inicial de a: " + a); // a vale 10  
    imprime_doble(a); // Se le pasa el 10 a la función  
    System.out.println("Valor final de a: " + a); // a sigue valiendo 10  
}
```

// El parámetro 'a' es independiente de la 'a' del main. ¡Son variables distintas!

```
public static void imprime_doble(int a) { // Se copia el valor 10 a esta nueva 'a'  
    a = 2 * a; // Se duplica el valor de la 'a' de esta función, no afecta fuera  
    System.out.println("Valor de a en la función: " + a); // 'a' vale 20  
}
```

Paso por referencia

Parámetros de tipo objeto (Como objetos de tipo String, los Arrays, etc.)

No se copia el objeto sino que se le pasa a la función una referencia al objeto original (un puntero). Por ello desde la función se accede directamente al objeto que se encuentra fuera. Los cambios que hagamos dentro de la función afectarán al objeto.

Paso por referencia

```
// Suma x a todos los elementos del vector v
public static void suma_x_al_vector(int v[], int x) {
    for (int i = 0; i < v.length; i++)
        v[i] = v[i] + x;
}

public static void main(String[] args) {
    int v[] = {0, 1, 2, 3};
    System.out.println("Vector antes: " + Arrays.toString(v));
    suma_x_al_vector(v, 10);
    System.out.println("Vector después: " + Arrays.toString(v));
}
```