

Together for Tomorrow!
Enabling People

Documentación técnica:

Funciones y programas básicos



1. Funciones.....	5
2. Programas Básicos.....	8

1. Funciones



Hay tres acciones elementales en programación: secuencia, selección y bucle.

Secuencia es una lista simple, consecutiva de acciones. El orden está implícito en el orden de las instrucciones individuales. Por ejemplo:

```
int contador=0; // inicialización
```

```
contador = contador +1;
```

```
System.out.println (contador); // escribe en pantalla el valor de contador
```

Las tres instrucciones se ejecutarán una tras otra empezando por la inicialización, luego el incremento de 1 en la variable contador y luego se escribirá en pantalla.

El mecanismo de **selección** básico es la sentencia **if-else**, que actúa según el valor de una condición. Se pueden encadenar varias sentencias de selección de dos formas distintas:

- o **Por concatenación**: una a continuación de la otra. Por ejemplo, una óptica hace un descuento del 10 % si el gasto por la compra de gafas es mayor que 100 euros. Si el cliente compra más de dos gafas, se le regala las fundas. El código en Java correspondiente sería algo así:

```
if (precio>100) {  
    precio=precio*0.9;  
}  
  
if (cantidad>2) {  
    regaloFundas;  
}
```

- o **Por anidamiento**: una dentro de otra. Siguiendo con el ejemplo anterior, la óptica hace un descuento del 10 % si el gasto por la compra de gafas es mayor que 100 euros. Pero si una de las gafas compradas es de sol, la tienda le añade otro 10% más de descuento, porque están en promoción. El código en Java correspondiente sería algo así:

1. Funciones



```
if (precio>100) {  
    precio=precio*0.9;  
    if (gafasSol==true){  
        precio=precio*0.9;  
    }  
}
```

En los lenguajes modernos existe una posibilidad de realizar selección múltiple. En función del valor de una variable seguiremos un camino u otro, el comportamiento es equivalente a una serie de if-anidados. Para ello, existe una sentencia especial que se llama **switch**. Su sintaxis es:

```
switch (expresion) {  
    case Valor1:  
        // Bloque sentencias 1  
        break;  
    case Valor2:  
        // Bloque sentencias 2  
        break;  
    case Valor3:  
        // Bloque sentencias 3  
        break;  
    ....  
    default:  
        // Bloque sentencias por defecto  
}
```

1. Funciones



Otra forma de alterar el flujo secuencial de instrucciones es utilizando bucles, los cuales permiten repetir las mismas instrucciones varias veces. En los programas es muy habitual encontrar estos bloques de **iteración o repetición**. El bucle más directo se implementa con la palabra **while**, que repite un bloque de sentencias mientras se mantiene cierta condición. Si la condición es falsa desde el principio, es decir, que no va a suceder desde el principio, no se ejecutará ninguna instrucción de todo el bloque contenido dentro del while. El otro tipo de bucle, el que comprueba la condición al final del bloque de sentencias, es el tipo **do-while**. Este tipo de bucle se ejecuta al menos una vez ya que la condición se ejecuta al final del bloque de sentencias.

Por último, se encuentran los bucles controlados por un contador. El bucle **for** se utiliza cuando se conoce el número de veces que se tiene que repetir el bucle. En un bucle for, se ejecuta el bloque de sentencias y luego se incrementa en uno el contador. Habitualmente se coloca en la cabecera del for los tres componentes necesarios sobre el contador: inicialización, límite para continuar en el bucle e incremento. Sin embargo, si falta alguna de ellas, el comportamiento por defecto es:

1. Si no se inicializa el contador, éste deberá haberse inicializado en el programa antes del bucle for.
2. Si no existe un valor límite para el contador, el bucle será infinito y no terminará nunca.
3. Si no se especifica un incremento, no se incrementará el contador.

2. Programas básicos



A continuación, se presentan una serie de ejemplos de programas básicos que utilizan las estructuras de acciones: secuencia, selección y bucle.

Un programa que escribe en pantalla una serie de números correspondientes a la **secuencia de Fibonacci**, que es en sí una sucesión matemática infinita. Se van sumando los números consecutivos de dos en dos, comenzando con 0 y 1 de la siguiente manera:

0,1,1,2,3,5,8,13,21,34...

(0, 1, 0+1=1, 1+1=2, 1+2=3, 2+3=5, 3+5=8, 5+8=13, 8+13=21, 13+21=34...)

Así sucesivamente, hasta el infinito. Utilizando un bucle for, el código en lenguaje Java podría ser:

```
// Programa escrito en lenguaje Java para escribir en pantalla la serie de Fibonacci
public static void main (String[] args) {
    // Declaración de variables
    int n1 =1;
    int n2;
    int max = 0; // guarda la cantidad de números que se van a escribir en pantalla
    int temp = 0; // variable auxiliar para guardar datos temporalmente
    // La variable leer guarda los datos introducidos por el usuario con el teclado
    Scanner leer = new Scanner (System.in);
    // Mensaje de pregunta al usuario
    System.out.println ("¿Cuántos números de la serie Fibonacci desea crear >2? ");
    // Guarda la cantidad de números de la serie que se van a escribir en pantalla      max      =
    leer.nextInt();
    // Un bucle para calcular y escribir los números de la secuencia de Fibonacci
    for (i=1; i<=max; i++){
        n2 = temp;
        temp = n1 + temp;
        n1 = n2;
        System.out.print (n1); System.out.print (",");
    }
} // fin del método main (principal)
```


2. Programas básicos



Un programa que controla las **ventas de gafas en una tienda de óptica**, que hace descuentos según el volumen de compra del cliente y le hace regalos si compra gafas de sol, sería el siguiente en código Java:

// Programa escrito en lenguaje Java para gestionar las ventas de la tienda de gafas

```
public static void main (String[] args) {  
    // Declaración de variables  
    boolean gafasSol=false;  
    int cantidad=0; // variable que guarda la cantidad de gafas que se van a comprar  
    int total = 0; // variable que guarda la cantidad de dinero a gastar  
    // La variable leer guarda los datos introducidos por el usuario con el teclado  
    Scanner leer = new Scanner (System.in);  
    // Mensaje de pregunta al usuario  
    System.out.println ("¿Cuántas gafas quiere comprar?");  
    // Guarda la cantidad de gafas que el cliente va a comprar  
    cantidad = leer.nextInt();  
    total = cantidad * 50; // cada par de gafas cuesta 50 euros  
    if (total>=100) {total = total *0.9;}  
    if (cantidad>=2) {regaloFundas;}  
    // Mensaje de pregunta al usuario  
    System.out.println ("¿Quiere unas gafas de sol? S=Si, N=no");  
    char caracter;  
    caracter = leer.next().charAt(0);  
    if (caracter == S) {gafasSol==true;}  
    if (gafasSol==true) {total = total *0.9;}  
    System.out.println ("Muchas gracias. Tiene que pagar: " + total);  
} // fin del método principal
```


SAMSUNG

BeJob[®]