





ÍNDICE

SENTENCIAS DE CONTROL

1. Sentencias de control de flujo	3
---	---



1. Sentencias de control de flujo

Sentencia condicional if

Existe una condición que se evalúa a cierto o falso; si resulta cierto, se evalúan las sentencias que están inmediatamente a continuación y si resulta falso se evalúan las sentencias dentro del else.

En Java, la condición evaluada por la instrucción if debe dar como resultado obligatoriamente, un tipo boolean, de lo contrario se producirá un error de compilación. Por otro lado, indicar que el bloque else es opcional

Ejemplo:

```
if(a>5){  
    System.out.println("aprobado");  
}  
else{  
    System.out.println("suspenso");  
}
```

Operador ternario

El operador ternario se emplea para asignar valores a una variable en función del cumplimiento o no de una condición, es decir, evalúa una condición y retorna un valor dependiendo si la condición es verdadera (true) o falsa (false).

La sintaxis del operador ternario es la siguiente:

pregunta ? trueresultado : falseresultado;

Ejemplo

Sentencias de control

```
int radio = (x<y)? a : b;
```

Que equivale a las sentencias con if:

```
if (x<y) { radio = a; }
```

```
else { radio = b; }
```

Sentencia condicional switch

El condicional switch, es usado para evaluar el valor de una variable y actuar de una forma u otra según sea su valor.

Ejemplo:

```
switch(nota){  
    case 5:  
    case 6:  
        System.out.println("aprobado");  
    case 7:  
    case 8:  
        System.out.println("notable");  
    case 9:  
    case 10:  
        System.out.println("sobresaliente");  
    default:  
        System.out.println("suspense");
```

Sentencias de control

```
}
```

Bucle for

Un bucle o ciclo, en programación, es una sentencia que se realiza repetidas veces a un trozo aislado de código hasta que la condición asignada a dicho bucle deje de cumplirse. El bucle FOR se utiliza para repetir más instrucciones, un determinado número de veces.

Ejemplo

Inicialización de las posiciones de un array a 0.

```
for (int i=0; i<10; i++) {  
    array[i]=0;  
    System.out.println("Posicion: " + i + " inicializada a 0");  
}
```

Bucle while

El bucle while se utiliza en la repetición condicional, un número de veces no determinado a priori. El bloque de sentencias puede no ejecutarse ninguna vez.

Ejemplo

```
x=1;  
  
while(x<5)  
{ x=x+1; System.out.println(x); }
```

Se imprimirá por pantalla: 2, 3, 4. Cuando x valga 5 se romperá el bucle y el flujo del programa saldrá de éste pasando a la siguiente instrucción al bucle while

Sentencias de control

Bucle Do/While

El bucle do/while se utiliza en la repetición condicional un número de veces no determinado a priori. El bloque de sentencias se ejecuta al menos una vez.

Ejemplo

```
x=1;
do{ System.out.println(x); x=x+1;
} while(x<5)
```

Se imprimirá por pantalla: 1, 2, 3, 4. Cuando x valga 5 se romperá el bucle y el flujo del programa saldrá de éste pasando a la siguiente instrucción al bucle while.

Salida forzada de un bucle

Consiste en usar la palabra reservada break para salirse de los bucles for/while/do por cumplirse cierta condición de break.

Además se puede utilizar el comando continue del mismo modo que el comando break sólo que en este caso se termina esta iteración y no el bucle completo.

Bucles etiquetados

Los bucles etiquetados son útiles para poner una etiqueta al bloque que queremos que se salte al utilizar break o continue.

La forma de etiquetar consiste en poner un nombre de etiqueta terminado con ":" exactamente antes del comienzo del bucle que queremos y añadiendo a la palabra clave break o continue el nombre de la etiqueta.

Ejemplo

```
salir:
for (int i=0; i<10; i++) {
while (x<50) { if (i*x == 400) break salir; sentencia2;
```

Sentencias de control

```
... sentencian;
```

```
}
```

```
sentencia2;
```

```
... sentencian;
```

```
}
```