ESTRUCTURAS DE CONTROL DE FLUJO

Introducción

Son aquellas sentencias que colocando una condición producen distintas alternativas.

Cada alternativa depende del resultado de la condición o expresión lógica.

Permite la ejecución condicional de fragmentos de código.

Bifurcación if

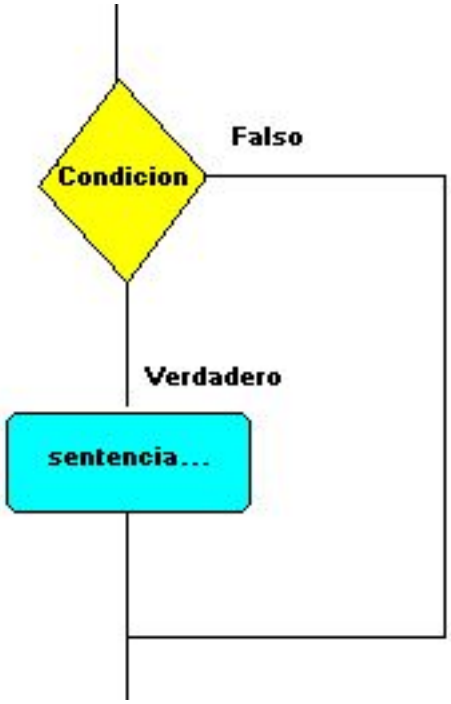
La estructura de control “if” permite decidir entre dos opciones resultantes de la evaluación de una sentencia. Si la evaluación es positiva se ejecuta una parte del código, de lo contrario el código dentro de la condición no se ejecuta. También podemos especificar acciones para realizar en caso de que la evaluación sea negativa.

Cuando se procesa un declaración “if” se evalúa la expresión de condición y el resultado es interpretado como un valor booleano. Si el resultado es verdadero, se ejecutan las sentencias contenidas dentro del “If”. Si el resultado es falso, simplemente se procede con la ejecución de la siguiente declaración.

Los condicionales pueden anidarse entre sí a una profundidad arbitraria.

Forma de uso:

[code]
if (expresionBooleana)
{
sentencias1;
}
[/code]



Bifurcación if else

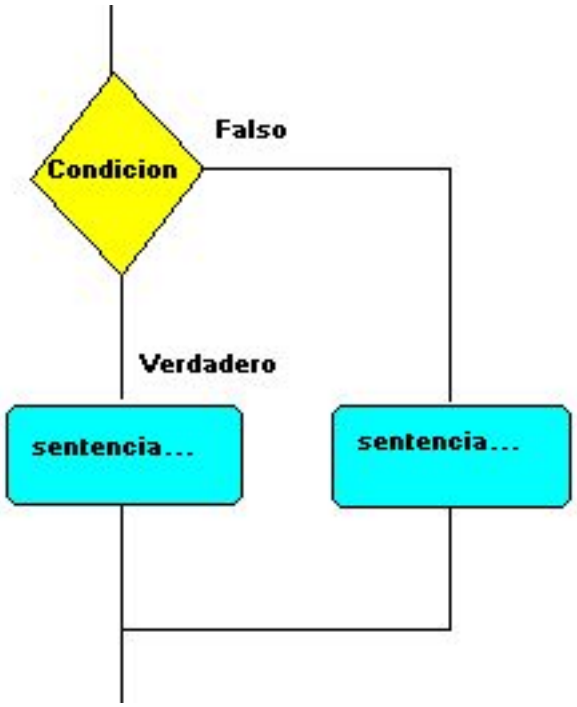
Es la extensión de la sentencia IF

Significa “de lo contrario” y permite la ejecución de un bloque de código si la condición de la sentencia IF fue falsa.

La sentencia ELSE se ejecuta solamente si la expresión IF se evalúa como falsa.

Forma de uso:

[code]
if (expresionBooleana)
{
sentencias1;
}
else
{
sentencias2;
}
[/code]



Bifurcación if else if else

Es una combinación entre if y else

Es una sentencia con su propia condición lógica en caso de que la sentencia IF anterior no haya sido válida.

Se pueden anidar sucesivos else if

Forma de uso:

[code]
if (expresionBooleana1) {
sentencias1;
} else if (expresionBooleana2) {
sentencias2;
} else if (expresionBooleana3) {
sentencias3;
} else {
sentencias4;
}
[/code]

Bifurcación switch

A diferencia de la estructura condicional “If”, la estructura “Switch” admite múltiples caminos a partir de la evaluación de una sola expresión.

La expresión puede ser una variable o cualquier otro tipo de expresión, siempre y cuando se evalúe un valor simple (es decir, un número entero, un doble, o una cadena). La construcción se ejecuta mediante la evaluación de la expresión y luego se prueba el resultado contra las sentencias **case**. Tan pronto como un valor coincidente se encuentra, las declaraciones posteriores se ejecutan en secuencia hasta la declaración break o hasta el final del Switch.

Su principal característica es que permite establecer muchas opciones según el valor de una variable (o expresión).
A diferencia de la sentencia IF que solo hay dos opciones (Verdadero – Falso).
Forma de uso:

[code]
switch (expresion)
{
case valor1: sentencias1; break;
case valor2: sentencias2; break;
case valor3: sentencias3; break;
default: sentencias4;
}
[/code]

Los valores no comprendidos en ninguna sentencia case se pueden gestionar en el default
Si no está el break, se ejecutan todas las sentencias case a continuación