



Estructuras de Control en C++

La declaración SWITCH

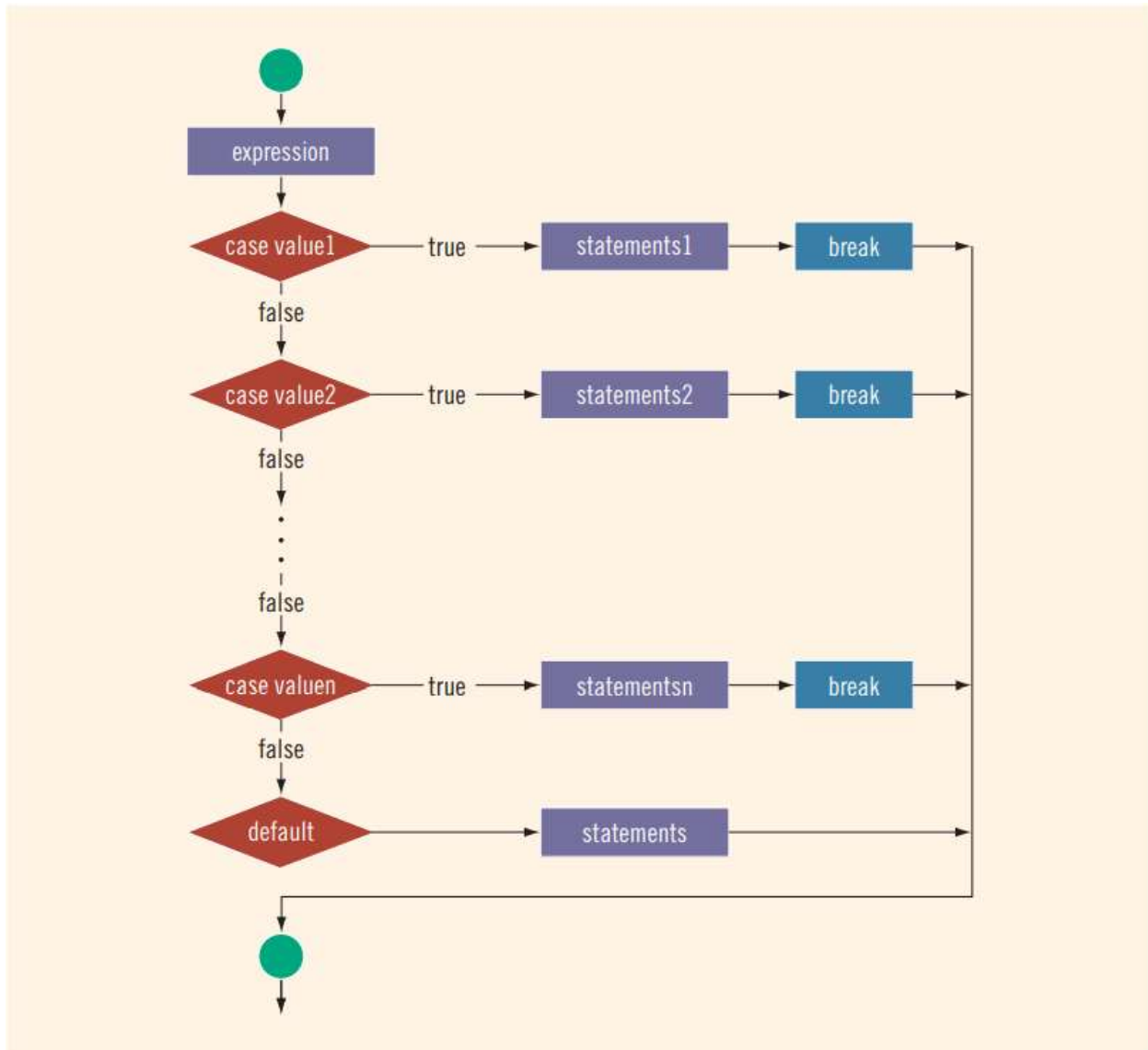
Recuerde que hay dos estructuras de selección en C++. La primera estructura, que se implementa con declaraciones `if` y `if ... else`, generalmente requiere la evaluación de una expresión (lógica). La segunda estructura de selección, que no requiere la evaluación de una expresión lógica, se llama ***switch***. El `switch` en C++ le da a la computadora el poder de elegir entre muchas alternativas.

La sintaxis general para la declaración del `switch` es:

```
1 switch(expression) {  
2     case x:  
3         // code block  
4         break;  
5     case y:  
6         // code block  
7         break;  
8     default:  
9         // code block  
10 }
```

En C ++, **switch**, **case**, **break** y **default** son palabras reservadas. Así es como funciona:

- La expresión del **switch** se evalúa una vez
- El valor de la expresión se compara con los valores de cada caso (**case**).
- Si hay una coincidencia, se ejecuta el bloque de código asociado
- Las palabras reservadas de **break** y **default** son opcionales (Dependiendo de la finalidad que le queramos dar a nuestro código).



break y default

break

Cuando C ++ alcanza una palabra break, sale del bloque switch. Esto detendrá la ejecución de más pruebas de código y casos dentro del bloque.

Cuando se encuentra una coincidencia y el trabajo está hecho, es hora de un break. No es necesario realizar más pruebas.

Un break puede ahorrar mucho tiempo de ejecución porque "ignora" la ejecución de todo el resto del código en el bloque switch.

default

La palabra default especifica algún código para ejecutar si no hay una coincidencia con la evaluación proporcionada al inicio del switch.

Nota: default debe usarse como la última declaración en el conmutador y no necesita un break.

switch vs if else

Una sentencia switch suele ser más eficiente que un conjunto de if anidados. Decidir si usar declaraciones if-else-if-else o una declaración switch se basa en la legibilidad y la expresión que la declaración está probando.

1. **Verificando la expresión de prueba:** una instrucción if-else-if-else puede probar expresiones basadas en rangos de valores o condiciones, mientras que una instrucción switch prueba expresiones basadas solo en un único entero, valor enumerado u objeto String.
2. **switch es mejor para ramificación de múltiples vías:** cuando el compilador compila un switch, inspeccionará cada una de las constantes de caso y creará una "tabla de salto" que utilizará para seleccionar la ruta de ejecución según el valor de la expresión. Por lo tanto, si necesitamos seleccionar entre un grupo grande de valores, una instrucción switch se ejecutará mucho más rápido que la lógica equivalente codificada usando una secuencia de if-elses. El compilador puede hacer esto porque sabe que las constantes de caso son todas del mismo tipo y simplemente deben compararse por igualdad con la expresión de cambio, mientras que en el caso de las expresiones if, el compilador no tiene tal conocimiento.
3. **if-else es mejor para valores booleanos:** las ramas condicionales If-else son excelentes para condiciones variables que dan como resultado un booleano, mientras que las declaraciones de cambio son excelentes para valores de datos fijos.
4. **Velocidad:** un switch puede resultar más rápida que ifs con la cantidad de casos verdaderos. Si hay pocos casos, es posible que no afecte la velocidad en ningún caso. Prefiere cambiar si el número de casos es más de 5; de lo contrario, también puede usar if-else. Si un switch contiene más de cinco elementos, se implementa mediante una tabla de búsqueda o una lista hash. Esto significa que todos los elementos obtienen el mismo tiempo de acceso, en comparación con una lista de ifs donde el último elemento tarda mucho más en llegar, ya que primero debe evaluar todas las condiciones anteriores.
5. **Claridad en la legibilidad:** un switch se ve mucho más limpio cuando tiene que combinar casos. Los ifs también son bastante vulnerables a errores. Perderse una declaración else puede causarle estragos. Agregar / quitar etiquetas también es más fácil con un switch y hace que su código sea mucho más fácil de cambiar y mantener.