



Introducción a la programación en C++

Depuración de un programa en C++

Los errores que se producen en los programas han de ser detectados, aislados (fijados) y corregidos. El proceso de encontrar errores se denomina depuración del programa. La corrección del error es probablemente la etapa más fácil, siendo la detección y aislamiento del error las tareas más difíciles.

Existen diferentes situaciones en las cuales se suelen introducir errores en un programa. Dos de las más frecuentes son:

1. Violación (no cumplimiento) de las reglas gramaticales del lenguaje de alto nivel en el que se escribe el programa.
2. Los errores en el diseño del algoritmo en el que está basado el programa.

Cuando el compilador detecta un error, visualiza un mensaje de error indicando que se ha cometido un error y posible causa del error. Desgraciadamente los mensajes de error son difíciles de interpretar y a veces se llegan a conclusiones erróneas. También varían de un compilador a otro compilador. A medida que se gana en experiencia, el proceso de puesta a punto de un programa se mejora considerablemente. Desde el punto de vista conceptual existen tres tipos de errores: **sintaxis**, **lógicos** y de **regresión**.

1. **Los errores de sintaxis** son aquellos que se producen cuando el programa viola la sintaxis, es decir, las reglas de gramática del lenguaje. Errores de sintaxis típicos son: escritura incorrecta de palabras reservadas, omisión de signos de puntuación (comillas, punto y coma...). Los errores de sintaxis son los más fáciles de fijar, ya que ellos son detectados y aislados por el compilador.
2. Un segundo tipo de error importante es **el error lógico**, ya que tal error representa errores del programador en el diseño del algoritmo y posterior programa. Los errores lógicos son más difíciles de encontrar y aislar, ya que no suelen ser detectados por el compilador.
3. **Los errores de regresión** son aquellos que se crean accidentalmente cuando se intenta corregir un error lógico. Siempre que se corrige un error se debe comprobar totalmente la exactitud (corrección) para asegurarse que se fija el error que se está tratando y no produce otro error. Los errores de regresión son comunes, pero son fáciles de leer y corregir. Una ley no escrita dice: «un error se ha producido, probablemente, por el último código modificado».

Errores en tiempo de ejecución:

Existen dos tipos de errores en tiempo de ejecución: aquellos que son detectados por el sistema en tiempo de ejecución del programa C++ —una vez traducido— y aquellos que permiten la terminación del programa, pero producen resultados incorrectos.

Un error en tiempo de ejecución puede tener como resultado que el programa obligue a la computadora a realizar una operación ilegal tal como dividir un número por cero o manipular datos no válidos o no definidos. Cuando ocurre este tipo de error, la computadora detendrá la ejecución de su programa y emitirá (visualizará) un mensaje de diagnóstico.

Mensajes de error:

Los compiladores emiten mensajes de error o de advertencia durante las fases de compilación, de enlace o de ejecución de un programa. Los mensajes de error producidos durante la compilación se suelen producir, normalmente, por errores de sintaxis y suele variar según los compiladores; pero, en general, se agrupan en tres bloques:

- **Errores fatales**, son raros. Algunos de ellos indican un error interno del compilador. Cuando ocurre un error fatal, la compilación se detiene inmediatamente, se debe tomar la acción apropiada y, a continuación, se vuelve a iniciar la compilación.
- **Errores de sintaxis**, son los errores típicos de sintaxis, errores de línea de órdenes y errores de acceso a memoria o disco. El compilador terminará la fase actual de compilación y se detiene.
- **Advertencias (warning)**, no impiden la compilación. Indican condiciones que son sospechosas, pero son legítimas como parte del lenguaje.

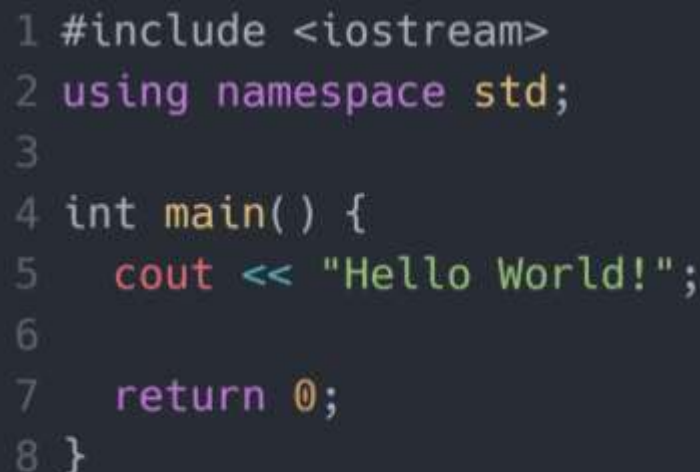
¿Qué es C++?

- C++ es un lenguaje multiplataforma que se puede utilizar para crear aplicaciones de alto rendimiento.
- C++ fue desarrollado por Bjarne Stroustrup, como una extensión del lenguaje C.
- C++ brinda a los programadores un alto nivel de control sobre los recursos del sistema y la memoria.
- El lenguaje se actualizó 3 veces importantes en 2011, 2014 y 2017 a C++ 11, C++ 14 y C++ 17.

¿Por qué usar C ++?

- C++ es uno de los lenguajes de programación más populares del mundo.
- C++ se puede encontrar en los sistemas operativos actuales, interfaces gráficas de usuario y sistemas integrados.
- C++ es un lenguaje de programación orientado a objetos que proporciona una estructura clara a los programas y permite reutilizar el código, lo que reduce los costos de desarrollo.
- C++ es portátil y se puede utilizar para desarrollar aplicaciones que se pueden adaptar a múltiples plataformas.
- ¡C++ es divertido y fácil de aprender!
- Como C++ está cerca de C # y Java, facilita a los programadores cambiar a C++ o viceversa.

Sintaxis de C++



```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     cout << "Hello World!";
6
7     return 0;
8 }
```

Línea 1: *#include <iostream>* es una biblioteca de archivos de encabezado que nos permite trabajar con objetos de entrada y salida, como *cout* (usado en la línea 5). Los archivos de encabezado agregan funcionalidad a los programas C++.

Línea 2: *using namespace std* significa que podemos usar nombres para objetos y variables de la biblioteca estándar.

Línea 4: Otra cosa que siempre aparece en un programa C++, es *int main()*. A esto se le llama función. Se ejecutará cualquier código dentro de sus llaves {}.

Línea 5: *cout* (pronunciado "see-out") es un objeto usado junto con el operador de inserción (<<) para generar/imprimir texto. En nuestro ejemplo, generará "Hello World!".

Línea 7: *return 0* finaliza la función principal.

Línea 8: No olvidemos agregar el corchete de cierre *}* para finalizar la función principal.

Nota: Cada instrucción de C++ termina con un punto y coma (;).