C++
Moderno

Guía rápida y concisa para programadores

Jorge Moreno Aguilera

C++ Moderno

Guía rápida y concisa para programadores

Jorge Moreno Aguilera

Este libro está a la venta en http://leanpub.com/cppmoderno

Esta versión se publicó en 2019-10-19

Índice general

C++ Moderno

Bienvenidos a C++ Moderno.

Este manual está pensado para programadores con conocimientos previos sobre lenguajes orientado a objetos: Java, Python, C#, etc. También puede ser muy útil para amantes y profesionales de C/C++ que necesiten una puesta a punto o simplemente refrescar conceptos. No recomiendo este texto para personas recién iniciadas o con poca experiencia, porque puede abordar temas complejos demasiado prematuramente.

Este libro nació como un intento de poner orden a distintas notas y apuntes que he ido redactando a lo largo de mi carrera profesional. Por ello, sin ser exactamente un manual de referencia, contiene mucha información de manera muy concentrada: si buscas ponerte al día o aprender algo nuevo (metaprogramación, *move-semantics*, etc.), ¡este libro puede serte de gran ayuda!

La flexibilidad de C++ es una de sus principales armas, por lo que podemos abordar un mismo problema de distintas formas. Claro está que no todas las formas son igual de convenientes, por lo que intentaremos plasmar de manera concisa una serie de directrices y buenas prácticas, que por seguro nos ahorrarán futuros dolores de cabeza.

Veamos a continuación de manera resumida qué temas trataremos en cada bloque de contenido:

En C++ un tipo cualquiera *T* viene en multitud de sabores: *T, T&, const T&, T*, T&&* y otras combinaciones exóticas. Parece complejo pero lo cierto es que hay una regla muy simple que los rige todos. Este contenido lo veremos en el capítulo sobre **value semantics**, junto a *la regla del cero* y *la regla del tres*: posiblemente las reglas más útiles y de uso más habitual.

En el capítulo de **move semantics** estudiaremos una nueva filosofía de trabajo: ahora los valores no solo se copian, ¡sino que también se mueven! Esta característica es muy poderosa, por lo que intentaremos profundizar en ella: veremos cómo funciona y cómo sacarle partido, ampliaremos la regla del tres a la regla del cinco, veremos algunos consejos de uso y añadiremos a nuestro arsenal el *idiom copy&swap*.

En el apartado de **metaprogramación** abordaremos una de las características más importantes del lenguaje. Aprenderemos todo lo relacionado con plantillas, profundizando en su diseño y las buenas prácticas en su uso.

Los **punteros** posiblemente sean una de las fuentes de quebraderos de cabeza más comunes entre los programadores de C++, por lo que dedicamos un capítulo a tratarlos en detalle para desmitificarlos. Hablaremos de los *smart pointers*, la *técnica RAII,* la gestión de memoria, etc.

En el bloque dedicado a **STL** veremos cómo acomodar tus propias clases para que funcionen con la librería estándar de C++. Existen más de un centenar de algoritmos útiles y eficientes que ya provee STL y que puedes usar en tus propias clases si son compatibles.

En el capítulo sobre **C++17 y C++20** veremos las características más novedosas del lenguaje, que nos permitirán escribir código más elegante y seguro.

A continuación veremos un bloque dedicado a las **buenas prácticas**, que hemos concebido como si fuera un recetario. Estas buenas prácticas han sido escogidas por su utilidad e impacto en el código.

En los últimos capítulos hablaremos sobre **patrones de diseño** (*singletons*, *observer*, etc.) **y algoritmia** (*partition scheme, quicksort, quickselect*, etc.), de cara a la aplicación real y puesta en práctica de todo lo visto a lo largo del manual.

¿Estás listo para un viaje rápido e intenso para conocer C++ moderno?

Gracias por leer.