



C++20-Highlights

Mit dem neuen Sprachstandard halten nützliche Werkzeuge wie Concepts, Ranges, Coroutinen und Module Einzug in C++. Die genannten Features zählen zu den Highlights von C++20.

ab Seite 6

C++20-Highlights

Die großen Vier

Concepts, Ranges, Coroutinen und Module erweitern C++

Concepts

Herkulesaufgabe: Zwei Extreme und die Rettung dank Concepts

Concepts als (un-)eingeschränkte Platzhalter und syntaktischer Zucker

Concepts definieren und wiederverwenden

Evolution oder Revolution? Entwicklung von Concepts

Ranges

Funktionskomposition mit der Ranges-Bibliothek

Coroutinen

Nebenläufige Systeme organisieren mit anhaltbaren Funktionen

Coroutinen implementieren mit cppcoro

Module

Unabhängige Ordnung: Die Vorteile von Modulen

C++20-Basics

Kernsprachen-Features

Unbekannte Highlights: Drei-Wege-Vergleichsoperator und Designated-Initialisierer

Bibliotheken

Erweiterung der Chrono-Bibliothek und zwei neue Datentypen

Nebenläufigkeit

Überblick zur Concurrency

Enumerationen

Scoped Enums in C++



ISO-Standardisierung

Wie geht die Standardisierung von C++?

84

6

Ausblick auf C++23

Ein Blick in die Proposals für künftige Neuerungen

92

12

Interview

C++-Erfinder Bjarne Stroustrup im Gespräch

100

18

24

Praxis & Methoden

32

Variadische Templates

Weniger Code dank Fold Expressions

106

36

Memory Leaks

Speicherlecks finden

112

42

Programmiertechniken

Metaprogrammierung mit dem Compiler Explorer

116

48

Multithreading-Programmierung mit HPX

120

56

SFINAE für Member Checker und bedingte Kompilierung anwenden

124

Design-Patterns

Das Builder-Pattern und Alternativen in C++

130

66

Tools

C++-Tricks

C++-Compiler im Überblick

136

70

Micro-Benchmarking mit der Google Benchmark Library

143

74

Sicheres C++

MISRA-C++: Regelwerke und Konformität

150

78

Werkzeugkasten

Prüfstand für Testwerkzeuge zur Codeanalyse

157

Sonstiges

Editorial

3

Impressum

133

C++20-Basics

C++20 hat auch unbekanntere Neuerungen zu bieten. Dazu zählen die neuen Features der Kernsprache, die Erweiterung der Chrono-Bibliothek sowie die Verbesserung der Concurrency. Und mit einem Blick in die Proposals lassen sich bereits jetzt Rückschlüsse auf C++23 ziehen.

ab Seite 66



Interview

Warum sollte man heute noch C++ lernen, und wie kann Neulingen der Einstieg gelingen? Welche neuen Features aus C++20 werden langfristig am meisten Bedeutung haben? Diese und weitere Fragen beantwortet C++-Erfinder Bjarne Stroustrup im Gespräch.

ab Seite 100

Praxis & Methoden

Auf das Basiswissen folgt die Umsetzung in den Arbeitsalltag: Mit weniger Code dank Fold Expressions, Methoden zum Aufspüren von Speicherlecks und Metaprogrammierung per Compiler Explorer tauchen Entwickler in die Praxis ein.

ab Seite 106



Tools

Entwickler sind wie Handwerker auf gute Werkzeuge angewiesen. Allerdings sollte man diese Hilfsmittel auch beherrschen. Es gilt, den Überblick zu behalten, ob bei der Vielzahl an Versionen verschiedener C++-Compiler oder beispielsweise den MISRA-C++-Regeln für sicheres Programmieren.

ab Seite 136