

Templates em C++

História do C++

Introdução

C++ é uma linguagem de programação de propósito geral que foi desenvolvida como uma extensão do C para incluir programação orientada a objetos e outras funcionalidades. Criada por Bjarne Stroustrup, a linguagem evoluiu significativamente ao longo das décadas desde sua concepção.

1. Origem e Desenvolvimento Inicial

- 1980: Bjarne Stroustrup começou a trabalhar em "C with Classes" enquanto estava nos Laboratórios Bell.
- 1983: A linguagem foi renomeada para C++ (pronunciado "C plus plus"), sugerindo a ideia de um incremento em relação ao C.

2. C++87

- 1987: A primeira versão comercial do C++ foi lançada. Esta versão incluiu classes, herança, funções virtuais, sobrecarga de operadores e funções, bem como a verificação de tipo forte.

3. C++98

- 1998: O primeiro padrão ISO do C++ foi lançado como ISO/IEC 14882:1998. Esta versão padronizou muitas funcionalidades que se tornaram centrais para a linguagem, incluindo a

Templates em C++

biblioteca padrão C++ (STL).

4. C++03

- 2003: Uma revisão menor do padrão C++98 foi lançada como ISO/IEC 14882:2003. Este padrão corrigiu bugs e fez pequenas melhorias, mas não introduziu novas funcionalidades significativas.

5. C++11

- 2011: Uma atualização significativa da linguagem foi lançada como ISO/IEC 14882:2011, também conhecida como C++0x antes de sua conclusão. Esta versão incluiu várias novas funcionalidades:

- Lambdas: Expressões lambda para definição de funções anônimas.
- Auto: Dedução automática de tipo.
- Smart Pointers: `std::shared_ptr`, `std::weak_ptr`.
- Move Semantics: Semântica de movimento e `std::move`.
- Threading: Biblioteca de threads.
- Range-based for: Loop for baseado em intervalo.

6. C++14

- 2014: ISO/IEC 14882:2014 trouxe melhorias e correções adicionais ao C++11:

- Relaxed Constraints: Relaxamento de restrições em expressões lambda.
- Binary Literals: Literais binários (0b prefixo).
- Generic Lambdas: Lambdas genéricos.

Templates em C++

7. C++17

- 2017: ISO/IEC 14882:2017 foi outra atualização importante:
 - `std::optional`: Representação de valores opcionais.
 - `std::variant`: Tipo variante seguro em tempo de compilação.
 - `std::any`: Contêiner de tipo seguro.
 - Parallel Algorithms: Algoritmos paralelos na STL.
 - Inline Variables: Variáveis inline.

8. C++20

- 2020: ISO/IEC 14882:2020 trouxe inovações significativas:
 - Concepts: Conceitos para programação genérica.
 - Ranges: Biblioteca de intervalos.
 - Coroutines: Suporte a corrotinas.
 - Modules: Módulos para melhor gestão de dependências.
 - Three-way Comparison: Operador de comparação (`<=>`).

9. C++23 e Além

- Em Desenvolvimento: O padrão C++23 está em desenvolvimento e trará ainda mais melhorias e funcionalidades, continuando a evolução da linguagem.

Dicas de Boas Práticas

Templates em C++

- Mantenha-se Atualizado: Compreender as versões mais recentes do C++ pode ajudar a aproveitar novas funcionalidades e melhorias de desempenho.
- Utilize a Biblioteca Padrão: Aproveite as ricas funcionalidades fornecidas pela STL e outras bibliotecas padrão para escrever código eficiente e limpo.
- Programação Moderna: Adote práticas de programação moderna e segura introduzidas em versões recentes do C++.

Esta seção abrange a história e evolução do C++. Para mais detalhes, consulte a documentação oficial: <https://en.cppreference.com/w/cpp/language/history>