Apêndice F

Precedência e associatividade no C++

A tabela seguinte sumariza as regras de precedência e associatividade dos operadores do C++. Operadores colocados na mesma secção têm a mesma precedência. As secções são separadas por § e apresentadas por ordem decrescente de precedência. Quanto à associatividade, apenas os operadores unários (com um único operando) e os operadores de atribuição se associam à direita: todos os outros associam-se à esquerda, como é habitual.

Descrição	**Sintaxe** (itálico: partes variáveis da sintaxe)
resolução de âmbito	*nome_de_classe* `::` *membro*
resolução de âmbito	*nome_de_espaço_nominativo* `::` *membro*
global	`::` *nome*
global	`::` *nome_qualificado*
§	
selecção de membro	*objecto* `.` *membro*
selecção de membro	*ponteiro* `->` *membro*
indexação	*ponteiro* `[` *expressão_inteira* `]`
invocação de função	*expressão* `(` *lista_expressões* `)`
construção de valor	*tipo* `(` *lista_expressões* `)`
incrementação sufixa	*lvalor* `++`
decrementação sufixa	*lvalor* `--`
identificação de tipo	`typeid (` *tipo* `)`
identificação de tipo durante a execução	`typeid (` *expressão* `)`
conversão verificada durante a execução	`dynamic_cast <` *tipo* `> (` *expressão* `)`
conversão verificada durante a compilação	`static_cast <` *tipo* `> (` *expressão* `)`
conversão não verificada (evitar)	`reinterpret_cast <` *tipo* `> (` *expressão* `)`
conversão constante (evitar)	`const_cast <` *tipo* `> (` *expressão* `)`
§	
tamanho de objecto	`sizeof` *expressão*
tamanho de tipo	`sizeof (` *tipo* `)`

incrementação prefixa	`++` *`lvalue`*
decrementação prefixa	`--` *`lvalue`*
negação *bit*-a-*bit* ou complemento para um	`compl` *`expressão`* (ou `~`)
negação	`not` *`expressão`*
simétrico	`-` *`expressão`*
identidade	`+` *`expressão`*
endereço de	`&` *`lvalue`*
conteúdo de	`*` *`expressão`*
construçãode variável dinâmica	`new` *`tipo`*
construçãode variável dinâmica com inicializadores	`new` *`tipo`* `(` *`lista_expressões`* `)`
construção localizada de variável dinâmica	`new (` *`lista_expressões`* `)` *`tipo`*
construção localizada de variável dinâmica com inicializadores	`new (` *`lista_expressões`* `)` *`tipo`* `(` *`lista_expressões`* `)`
destruição de variável dinâmica	`delete` *`ponteiro`*
destruição de matriz dinâmica	`delete[]` *`ponteiro`*
conversão de tipo de baixo nível (crime!)	`(` *`tipo`* `)` *`expressão`*
§	
selecção de membro	*`objecto`* `.*` *`ponteiro_para_membro`*
selecção de membro	*`ponteiro`* `->*` *`ponteiro_para_membro`*
§	
multiplicação	*`expressão`* `*` *`expressão`*
divisão	*`expressão`* `/` *`expressão`*
resto da divisão inteira	*`expressão`* `%` *`expressão`*
§	
adição	*`expressão`* `+` *`expressão`*
subtracção	*`expressão`* `-` *`expressão`*
§	
deslocamento para a esquerda	*`expressão`* `<<` *`expressão`*
deslocamento para a direita	*`expressão`* `>>` *`expressão`*
§	
menor	*`expressão`* `<` *`expressão`*
menor ou igual	*`expressão`* `<=` *`expressão`*
maior	*`expressão`* `>` *`expressão`*
maior ou igual	*`expressão`* `>=` *`expressão`*
§	
igual	*`expressão`* `==` *`expressão`*
diferente	*`expressão`* `!=` *`expressão`* (ou `not_eq`)
§	
conjunção *bit*-a-*bit*	*`expressão`* `bitand` *`expressão`* (ou `&`)
§	
disjunção exclusiva *bit*-a-*bit*	*`expressão`* `xor` *`expressão`* (ou `^`)
§	
disjunção *bit*-a-*bit*	*`expressão`* `bitor` *`expressão`* (ou `|`)

	§
conjunção	*expressão* `and` *expressão* (ou `&&`)
	§
disjunção	*expressão* `or` *expressão* (ou `||`)
	§
operador condicional	*expressão* `?` *expressão* `:` *expressão*
	§
atribuição simples	*lvalue* `=` *expressão*
multiplicação e atribuição	*lvalue* `*=` *expressão*
divisão e atribuição	*lvalue* `/=` *expressão*
resto e atribuição	*lvalue* `%=` *expressão*
adição e atribuição	*lvalue* `+=` *expressão*
subtracção e atribuição	*lvalue* `-=` *expressão*
deslocamento para a esquerda e atribuição	*lvalue* `<<=` *expressão*
deslocamento para a direita e atribuição	*lvalue* `>>=` *expressão*
conjunção *bit*-a-*bit* e atribuição	*lvalue* `&=` *expressão* (ou `and_eq`)
disjunção *bit*-a-*bit* e atribuição	*lvalue* `|=` *expressão* (ou `or_eq`)
disjunção exclusiva *bit*-a-*bit* e atribuição	*lvalue* `^=` *expressão* (ou `xor_eq`)
	§
lançamento de excepção	`throw` *expressão*
sequenciamento	*expressão* `,` *expressão*