

Templates em C++

Expressões de Dobra em C++

Introdução

Expressões de dobra (fold expressions) foram introduzidas no C++17 para simplificar a manipulação de pacotes de parâmetros variáveis em templates. Elas permitem aplicar um operador a todos os elementos de um pacote de parâmetros, tornando o código mais conciso e legível.

1. Definição e Sintaxe

- Definição: Expressões de dobra aplicam um operador a todos os elementos de um pacote de parâmetros.

- Sintaxe:

- Dobra unária:

- (... op pack) // Direita

- (pack op ...) // Esquerda

- Dobra binária:

- (init op ... op pack) // Direita

- (pack op ... op init) // Esquerda

2. Exemplo Básico de Dobra Unária Direita

- Definição: Uma dobra unária direita aplica o operador a partir do lado direito do pacote de

Templates em C++

parâmetros.

- Exemplo:

```
template <typename... Args>

auto soma(Args... args) {

    return (... + args); // Soma todos os argumentos

}

int main() {

    std::cout << soma(1, 2, 3, 4) << std::endl; // Saída: 10

    return 0;

}
```

3. Exemplo Básico de Dobra Unária Esquerda

- Definição: Uma dobra unária esquerda aplica o operador a partir do lado esquerdo do pacote de parâmetros.

- Exemplo:

```
template <typename... Args>

auto produto(Args... args) {

    return (args * ...); // Multiplica todos os argumentos

}

int main() {

    std::cout << produto(1, 2, 3, 4) << std::endl; // Saída: 24

    return 0;

}
```

Templates em C++

```
}
```

4. Exemplo de Dobra Binária Direita

- Definição: Uma dobra binária direita aplica o operador a partir do lado direito do pacote de parâmetros, usando um valor inicial.

- Exemplo:

```
template <typename... Args>
auto somaComInicial(Args... args) {
    return (0 + ... + args); // Soma todos os argumentos começando com 0
}
```

```
int main() {
    std::cout << somaComInicial(1, 2, 3, 4) << std::endl; // Saída: 10
    return 0;
}
```

5. Exemplo de Dobra Binária Esquerda

- Definição: Uma dobra binária esquerda aplica o operador a partir do lado esquerdo do pacote de parâmetros, usando um valor inicial.

- Exemplo:

```
template <typename... Args>
auto produtoComInicial(Args... args) {
    return (args * ... * 1); // Multiplica todos os argumentos começando com 1
}
```

Templates em C++

```
}
```

```
int main() {  
    std::cout << produtoComInicial(1, 2, 3, 4) << std::endl; // Saída: 24  
    return 0;  
}
```

6. Uso com Operadores Lógicos

- Definição: Expressões de dobra podem ser usadas com operadores lógicos para verificar condições em todos os elementos de um pacote de parâmetros.

- Exemplo:

```
template <typename... Args>  
bool todosVerdadeiros(Args... args) {  
    return (... && args); // Verifica se todos os argumentos são verdadeiros  
}
```

```
int main() {  
    std::cout << std::boolalpha;  
    std::cout << todosVerdadeiros(true, true, true) << std::endl; // Saída: true  
    std::cout << todosVerdadeiros(true, false, true) << std::endl; // Saída: false  
    return 0;  
}
```

7. Uso com Operadores de Comparação

Templates em C++

- Definição: Expressões de dobra podem ser usadas com operadores de comparação para verificar condições de ordenação em todos os elementos de um pacote de parâmetros.

- Exemplo:

```
template <typename... Args>

bool todosMenores(Args... args) {

    return (... < args); // Verifica se todos os argumentos são menores que os seguintes
}

int main() {

    std::cout << std::boolalpha;

    std::cout << todosMenores(1, 2, 3, 4) << std::endl; // Saída: true

    std::cout << todosMenores(4, 3, 2, 1) << std::endl; // Saída: false

    return 0;

}
```

Dicas de Boas Práticas

- Simplificação de Código: Use expressões de dobra para simplificar o código que opera com pacotes de parâmetros.
- Clareza e Manutenção: Mantenha o uso de expressões de dobra claro e bem documentado para facilitar a leitura e a manutenção do código.
- Verificação de Tipos: Verifique se os tipos dos parâmetros são compatíveis com o operador usado na expressão de dobra para evitar erros de compilação ou execução.

Templates em C++

Esta seção abrange os conceitos sobre expressões de dobra em C++. Para mais detalhes, consulte a documentação oficial: <https://en.cppreference.com/w/cpp/language/fold>