

Templates em C++

Dedução de Argumento de Template em C++

Introdução

A dedução de argumento de template em C++ permite que o compilador deduza automaticamente os tipos de argumentos de template com base nos argumentos passados para a função. Isso simplifica a chamada de funções template, tornando o código mais legível e fácil de escrever.

1. Definição e Sintaxe

- Definição: A dedução de argumento de template é o processo pelo qual o compilador determina automaticamente os tipos dos parâmetros de template a partir dos argumentos fornecidos na chamada da função.

- Sintaxe:

```
template <typename T>
```

```
void nomeFuncao(T argumento);
```

```
// Chamada da função sem especificar o tipo T explicitamente
```

```
nomeFuncao(valor);
```

2. Exemplo Básico de Dedução de Argumento de Template

- Definição: Um exemplo básico que demonstra a dedução automática do tipo de argumento de template.

Templates em C++

- Exemplo:

```
template <typename T>

void imprimir(T valor) {

    std::cout << valor << std::endl;

}

int main() {

    imprimir(42);      // Deduzido como int

    imprimir(3.14);    // Deduzido como double

    imprimir("texto"); // Deduzido como const char*

    return 0;

}
```

3. Dedução com Múltiplos Argumentos de Template

- Definição: A dedução de argumento de template pode ser aplicada a funções template com múltiplos parâmetros de template.

- Exemplo:

```
template <typename T, typename U>

void imprimir(T primeiro, U segundo) {

    std::cout << primeiro << " " << segundo << std::endl;

}

int main() {

    imprimir(42, 3.14); // Deduzido como int e double

}
```

Templates em C++

```
imprimir("texto", 100);    // Deduzido como const char* e int

return 0;

}
```

4. Dedução com Parâmetros de Template Padrão

- Definição: Parâmetros de template podem ter valores padrão que são usados se nenhum argumento for fornecido.

- Exemplo:

```
template <typename T = int>

void imprimir(T valor = 0) {

    std::cout << valor << std::endl;

}
```

```
int main() {

    imprimir();           // Deduzido como int, valor padrão 0

    imprimir(3.14);       // Deduzido como double

    return 0;

}
```

5. Dedução com Funções Template Sobrecargas

- Definição: A dedução de argumento de template funciona com funções template sobrecarregadas.

- Exemplo:

```
template <typename T>
```

Templates em C++

```
void imprimir(T valor) {  
    std::cout << valor << std::endl;  
}
```

```
template <typename T, typename U>  
void imprimir(T primeiro, U segundo) {  
    std::cout << primeiro << " " << segundo << std::endl;  
}
```

```
int main() {  
    imprimir(42);           // Chama a versão com um argumento  
    imprimir(42, 3.14);     // Chama a versão com dois argumentos  
    return 0;  
}
```

6. Dedução de Template com Referências e Ponteiros

- Definição: A dedução de argumento de template pode ser aplicada a referências e ponteiros.

- Exemplo:

```
template <typename T>  
void imprimir(T& valor) {  
    std::cout << valor << std::endl;  
}
```

```
template <typename T>
```

Templates em C++

```
void imprimir(T* valor) {  
    std::cout << *valor << std::endl;  
}  
  
int main() {  
    int x = 42;  
    imprimir(x);      // Deduzido como int&  
    imprimir(&x);     // Deduzido como int*  
    return 0;  
}
```

7. Dedução de Argumento de Template com Tipos de Retorno

- Definição: A dedução de argumento de template também pode ser aplicada a tipos de retorno.
- Exemplo:

```
template <typename T>  
auto quadrado(T valor) -> decltype(valor * valor) {  
    return valor * valor;  
}  
  
int main() {  
    std::cout << quadrado(5) << std::endl;    // Deduzido como int  
    std::cout << quadrado(3.14) << std::endl; // Deduzido como double  
    return 0;  
}
```

Templates em C++

Dicas de Boas Práticas

- Simplificação de Código: Use a dedução de argumento de template para simplificar a chamada de funções template.
- Clareza e Manutenção: Mantenha a dedução de argumento de template clara e bem documentada para facilitar a leitura e a manutenção do código.
- Verificação de Tipos: Verifique se os tipos deduzidos são os esperados para evitar erros de compilação ou execução.

Esta seção abrange os conceitos sobre dedução de argumento de template em C++. Para mais detalhes, consulte a documentação oficial:
https://en.cppreference.com/w/cpp/language/template_argument_deduction