

Templates em C++

Dedução de Argumento de Template de Classe em C++

Introdução

A dedução de argumento de template de classe, introduzida no C++17, permite que o compilador deduza automaticamente os tipos de argumentos de template ao instanciar um objeto de uma classe template. Isso simplifica a criação de objetos de classes template, tornando o código mais legível e fácil de escrever.

1. Definição e Sintaxe

- Definição: A dedução de argumento de template de classe é o processo pelo qual o compilador determina automaticamente os tipos dos parâmetros de template a partir dos argumentos fornecidos ao construir um objeto de uma classe template.

- Sintaxe:

```
template <typename T>
```

```
class NomeClasse {
```

```
public:
```

```
    NomeClasse(T argumento);
```

```
};
```

```
// Instanciação da classe sem especificar o tipo T explicitamente
```

```
NomeClasse objeto(valor);
```

Templates em C++

2. Exemplo Básico de Dedução de Argumento de Template de Classe

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

- Exemplo:

```
template <typename T>
class Caixa {
public:
    T valor;

    Caixa(T v) : valor(v) {}
};

int main() {
    Caixa caixaInt(10);    // Deduzido como Caixa<int>
    Caixa caixaDouble(3.14); // Deduzido como Caixa<double>
    return 0;
}
```

3. Dedução com Múltiplos Argumentos de Template

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

- Exemplo:

```
template <typename T, typename U>
```

Templates em C++

```
class Par {  
  
public:  
  
    T primeiro;  
  
    U segundo;  
  
    Par(T p, U s) : primeiro(p), segundo(s) {}  
  
};  
  
int main() {  
  
    Par par(42, 3.14);      // Deduzido como Par<int, double>  
  
    Par par2("texto", 100); // Deduzido como Par<const char*, int>  
  
    return 0;  
  
}
```

4. Dedução com Construtores Template

- Definição: Construtores template permitem deduzir tipos de template ao instanciar objetos de classes template.

- Exemplo:

```
template <typename T>  
  
class Exemplo {  
  
public:  
  
    T valor;  
  
    template <typename U>
```

Templates em C++

```
Exemplo(U v) : valor(v) {}

};

int main() {

    Exemplo ex(42);      // Deduzido como Exemplo<int>

    Exemplo ex2(3.14);   // Deduzido como Exemplo<double>

    Exemplo ex3("texto"); // Deduzido como Exemplo<const char*>

    return 0;

}
```

5. Dedução com Agregados

- Definição: A dedução de argumento de template de classe pode ser usada com classes agregadas.

- Exemplo:

```
template <typename T>

struct Agregado {

    T valor;

};

int main() {

    Agregado agrInt{10};      // Deduzido como Agregado<int>

    Agregado agrDouble{3.14}; // Deduzido como Agregado<double>

    return 0;

}
```

Templates em C++

6. 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>
```

```
class Valor {
```

```
public:
```

```
    T valor;
```

```
    Valor(T v = 0) : valor(v) {}
```

```
};
```

```
int main() {
```

```
    Valor valor1;    // Deduzido como Valor<int>
```

```
    Valor valor2(3.14); // Deduzido como Valor<double>
```

```
    return 0;
```

```
}
```

7. Dedução de Template com Tipos de Referência e Ponteiros

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

- Exemplo:

Templates em C++

```
template <typename T>

class Referencia {

public:

    T& ref;


    Referencia(T& r) : ref(r) {}

};


int main() {

    int x = 42;

    Referencia ref(x);    // Deduzido como Referencia<int>

    return 0;

}
```

Dicas de Boas Práticas

- Simplificação de Código: Use a dedução de argumento de template de classe para simplificar a criação de objetos de classes template.
- Clareza e Manutenção: Mantenha a dedução de argumento de template de classe 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 de classe em C++.

Para mais detalhes, consulte a documentação oficial:

Templates em C++

https://en.cppreference.com/w/cpp/language/class_template_argument_deduction