

Templates em C++

Template de Variável em C++

Introdução

Templates de variável, introduzidos no C++14, permitem a criação de variáveis genéricas que podem ser instanciadas com diferentes tipos de dados. Eles são úteis para definir constantes e variáveis genéricas de forma flexível.

1. Definição e Sintaxe

- Definição: Um template de variável é um modelo de variável que pode ser instanciado com diferentes tipos de dados.

- Sintaxe:

```
template <typename T>
```

```
T nomeVariavel;
```

2. Exemplo Básico de Template de Variável

- Definição: Um template de variável básico que define uma constante PI genérica.

- Exemplo:

```
template <typename T>
```

```
constexpr T pi = T(3.1415926535897932385);
```

```
int main() {
```

Templates em C++

```
std::cout << pi<double> << std::endl; // Uso do template de variável com double

std::cout << pi<float> << std::endl; // Uso do template de variável com float

return 0;

}
```

3. Template de Variável com Parâmetros Não-Tipo

- Definição: Templates de variável podem ter parâmetros de template que não são tipos.
- Exemplo:

```
template <typename T, int N>

constexpr T arraySum(const T (&arr)[N]) {

    T sum = 0;

    for (int i = 0; i < N; ++i) {

        sum += arr[i];

    }

    return sum;

}
```

```
int main() {

    int arr[] = {1, 2, 3, 4, 5};

    std::cout << arraySum(arr) << std::endl; // Uso do template de variável com parâmetro não-tipo

    return 0;

}
```

4. Template de Variável Especializado

Templates em C++

- Definição: Templates de variável podem ser especializados para tipos específicos.

- Exemplo:

```
template <typename T>
```

```
constexpr T valorMaximo = T(100);
```

```
template <>
```

```
constexpr int valorMaximo<int> = 200;
```

```
int main() {
```

```
    std::cout << valorMaximo<double> << std::endl; // Uso da especialização padrão
```

```
    std::cout << valorMaximo<int> << std::endl;    // Uso da especialização para int
```

```
    return 0;
```

```
}
```

5. Uso de Template de Variável para Aliases

- Definição: Templates de variável podem ser usados para criar aliases de variáveis constantes.

- Exemplo:

```
template <typename T>
```

```
constexpr T e = T(2.71828182845904523536);
```

```
int main() {
```

```
    std::cout << e<double> << std::endl; // Uso do template de variável com double
```

```
    std::cout << e<float> << std::endl; // Uso do template de variável com float
```

Templates em C++

```
    return 0;  
}
```

6. Combinação de Templates de Variável com Outros Templates

- Definição: Templates de variável podem ser combinados com templates de função ou classe para criar soluções genéricas mais complexas.

- Exemplo:

```
template <typename T>  
constexpr T pi = T(3.1415926535897932385);
```

```
template <typename T>  
T calcularAreaCirculo(T raio) {  
    return pi<T> * raio * raio;  
}
```

```
int main() {  
    std::cout << calcularAreaCirculo(5.0) << std::endl; // Uso combinado de template de variável e  
função  
    return 0;  
}
```

Dicas de Boas Práticas

- Reutilização de Código: Use templates de variável para criar constantes genéricas reutilizáveis.

Templates em C++

- Especialização: Especialize templates de variável quando necessário para lidar com tipos específicos.
- Clareza e Manutenção: Mantenha os templates de variável claros e bem documentados para facilitar a manutenção do código.

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