Templates em C++

Pacote de Parâmetros em C++

Introdução

Os pacotes de parâmetros (parameter packs) em C++ permitem que você trabalhe com um número variável de argumentos de template. Eles são particularmente úteis para criar funções e classes genéricas que podem aceitar qualquer quantidade de parâmetros.

1. Definição e Sintaxe

- Definição: Um pacote de parâmetros é uma maneira de representar um número arbitrário de parâmetros de template.
- Sintaxe:

```
template <typename... Args>
void funcaoVariadica(Args... args) {
    // Implementação
}
```

2. Exemplo Básico de Função com Pacote de Parâmetros

- Definição: Uma função template que aceita um número variável de argumentos.
- Exemplo:

```
template <typename... Args>
void imprimir(Args... args) {
```

```
    (std::cout << ... << args) << std::endl;
}

int main() {
    imprimir(1, 2, 3);              // Saída: 123
    imprimir("Hello", " ", "World"); // Saída: Hello World
    return 0;
}
```

3. Expansão de Pacotes de Parâmetros

- Definição: A expansão de pacotes de parâmetros é o processo pelo qual os argumentos do pacote são descompactados.
- Exemplo:

```
template <typename... Args>
void imprimir(Args... args) {
    (std::cout << ... << args) << std::endl;
}

template <typename T>
void chamadaVariadica(T arg) {
    std::cout << arg << std::endl;
}

template <typename T, typename... Args>
```

```
 void chamadaVariadica(T primeiro, Args... restantes) {
     std::cout << primeiro << std::endl;
     chamadaVariadica(restantes...);
 }

 int main() {
     chamadaVariadica(1, 2, 3); // Saída: 1
2
3
     return 0;
 }
```

4. Pacotes de Parâmetros em Classes Template

- Definição: Pacotes de parâmetros também podem ser usados em classes template para aceitar um número variável de argumentos de template.
- Exemplo:

```
 template <typename... Args>
 class Exemplo {
 public:
     Exemplo(Args... args) {
         (std::cout << ... << args) << std::endl;
     }
 };
```

```
int main() {
    Exemplo<int, double, std::string> ex(1, 3.14, "texto"); // Saída: 1 3.14 texto
    return 0;
}
```

5. Pacotes de Parâmetros em Construtores

- Definição: Construtores de classes template podem utilizar pacotes de parâmetros para aceitar um número variável de argumentos.
- Exemplo:

```
template <typename... Args>
class Variadica {
public:
    Variadica(Args... args) {
        (std::cout << ... << args) << std::endl;
    }
};

int main() {
    Variadica<int, double, std::string> var(1, 3.14, "texto"); // Saída: 1 3.14 texto
    return 0;
}
```

6. Pacotes de Parâmetros em Herança

Templates em C++

- Definição: Pacotes de parâmetros podem ser usados em herança para passar um número variável de argumentos para a classe base.

- Exemplo:

```
template <typename... Args>
class Base {
public:
    Base(Args... args) {
        (std::cout << ... << args) << std::endl;
    }
};

template <typename... Args>
class Derivada : public Base<Args...> {
public:
    Derivada(Args... args) : Base<Args...>(args...) {}
};

int main() {
    Derivada<int, double, std::string> der(1, 3.14, "texto"); // Saída: 1 3.14 texto
    return 0;
}
```

7. Recursão com Pacotes de Parâmetros

- Definição: Pacotes de parâmetros podem ser processados recursivamente para implementar

lógica complexa.

- Exemplo:

```
 template <typename T>
 void imprimirRecursivo(T arg) {
     std::cout << arg << std::endl;
 }

 template <typename T, typename... Args>
 void imprimirRecursivo(T primeiro, Args... restantes) {
     std::cout << primeiro << std::endl;
     imprimirRecursivo(restantes...);
 }

 int main() {
     imprimirRecursivo(1, 2, 3); // Saída: 1
2
3
     return 0;
 }
```

Dicas de Boas Práticas

- Reutilização de Código: Use pacotes de parâmetros para criar funções e classes que possam aceitar um número variável de argumentos.
- Clareza e Manutenção: Mantenha a lógica de expansão de pacotes de parâmetros clara e bem

documentada para facilitar a leitura e a manutenção do código.

- Verificação de Tipos: Verifique se os tipos dos parâmetros são os esperados para evitar erros de compilação ou execução.

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