

# Conceitos Básicos de C++

## Pesquisa de Nome em C++

### Introdução

A pesquisa de nome em C++ refere-se ao processo pelo qual o compilador resolve os nomes das entidades (variáveis, funções, classes, etc.) no código. Esse processo é essencial para garantir que o código seja compilado corretamente, associando cada nome ao seu significado apropriado.

### 1. Escopo

O escopo define a região do programa onde um nome é válido. C++ possui diferentes tipos de escopo:

- Escopo de Bloco: Dentro de um par de chaves {}.

```
{  
    int x = 10; // x tem escopo limitado a este bloco  
}
```

- Escopo de Função: Dentro do corpo de uma função.

```
void funcao() {  
    int y = 20; // y tem escopo limitado a esta função  
}
```

- Escopo de Arquivo: Fora de todas as funções e classes, visível em todo o arquivo.

```
int z = 30; // z tem escopo de arquivo
```

## Conceitos Básicos de C++

- Escopo de Classe: Dentro da definição de uma classe.

```
class Exemplo {  
  
    int a; // a tem escopo de classe  
  
};
```

### 2. Pesquisa de Nome em Escopo de Bloco

Nomes são resolvidos no escopo mais próximo primeiro, movendo-se para fora em direção aos escopos externos se não forem encontrados.

Exemplo:

```
int x = 5;  
  
void funcao() {  
  
    int x = 10; // x local sobrescreve x global  
  
    // usa x = 10  
  
}  
  
// usa x = 5
```

### 3. Pesquisa de Nome em Espaços de Nomes

Namespaces ajudam a evitar conflitos de nome organizando o código em espaços de nomes distintos.

Exemplo:

## Conceitos Básicos de C++

```
namespace Espaco1 {  
    int valor = 100;  
}
```

```
namespace Espaco2 {  
    int valor = 200;  
}
```

```
int main() {  
    int a = Espaco1::valor; // usa valor de Espaco1  
    int b = Espaco2::valor; // usa valor de Espaco2  
}
```

### 4. Pesquisa de Nome em Classes e Herança

Nomes de membros de classe são resolvidos dentro do escopo da classe e suas classes base, se houver.

Exemplo:

```
class Base {  
    public:  
        int valor;  
};
```

```
class Derivada : public Base {
```

## Conceitos Básicos de C++

public:

```
void funcao() {  
    valor = 10; // usa valor da classe Base  
}  
};
```

### 5. Pesquisa de Nome em Templates

Em templates, a pesquisa de nome pode ser feita no escopo do template ou no escopo do instanciamento.

Exemplo:

```
template <typename T>  
void funcaoTemplate(T valor) {  
    T x = valor; // T é resolvido durante a instanciamento  
}
```

### 6. Pesquisa de Nome com using

A palavra-chave using pode ser usada para trazer nomes de um namespace ou classe base para o escopo atual.

Exemplo:

```
namespace MeuEspaco {  
    int valor = 50;
```

## Conceitos Básicos de C++

```
}
```

```
using MeuEspaco::valor;
```

```
int main() {
```

```
    int a = valor; // usa valor de MeuEspaco
```

```
}
```

### Dicas de Boas Práticas

- Evitar Conflitos: Use namespaces para evitar conflitos de nomes em projetos grandes.
- Clareza: Prefira nomes descritivos para variáveis e funções para facilitar a leitura e manutenção do código.
- Consistência: Mantenha uma nomenclatura consistente para aumentar a previsibilidade do código.
- Documentação: Documente o uso de namespaces e escopos complexos para facilitar a compreensão.

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