

# Elisão de Cópia em C++

## Introdução

A elisão de cópia em C++ refere-se a uma otimização do compilador que elimina cópias desnecessárias de objetos, tornando o código mais eficiente. Esta otimização é permitida e, em alguns casos, obrigatória, proporcionando melhorias significativas no desempenho.

### 1. Definição e Sintaxe

- Definição: A elisão de cópia elimina a criação e destruição temporária de objetos, evitando cópias desnecessárias.
- Sintaxe: A elisão de cópia ocorre de forma implícita e não requer sintaxe específica, sendo aplicada pelo compilador em contextos apropriados.

### 2. Exemplo de Elisão de Cópia

- Exemplo:

```
class Exemplo {  
  
public:  
  
    Exemplo() {}  
  
    Exemplo(const Exemplo&) {  
        std::cout << "Cópia realizada" << std::endl;  
    }  
};
```

```
Exemplo criarExemplo() {  
    return Exemplo(); // Elisão de cópia pode ocorrer aqui  
}
```

```
int main() {  
    Exemplo e = criarExemplo(); // Elisão de cópia pode ocorrer aqui  
    return 0;  
}
```

### 3. Elisão de Cópia em Inicialização de Variáveis

- Definição: A elisão de cópia pode ocorrer quando uma variável é inicializada com o valor de retorno de uma função que retorna um objeto por valor.

- Exemplo:

```
Exemplo criarExemplo() {  
    return Exemplo();  
}
```

```
int main() {  
    Exemplo e = criarExemplo(); // Elisão de cópia pode ocorrer aqui  
    return 0;  
}
```

### 4. Elisão de Cópia em Inicialização de Membros

- Definição: A elisão de cópia pode ocorrer durante a inicialização de membros em listas de

inicialização de construtores.

- Exemplo:

```
class Contenedor {  
  
public:  
  
    Exemplo e;  
  
    Contenedor() : e(Exemplo()) {} // Elisão de cópia pode ocorrer aqui  
  
};
```

## 5. Regras e Comportamentos Especiais

- Elisão de Cópia Obrigatória: Desde o C++17, a elisão de cópia é obrigatória em determinados contextos, como na inicialização de uma variável com o valor de retorno de uma função.
- RVO (Return Value Optimization): Otimização de valor de retorno, onde o compilador elimina a cópia do objeto retornado por valor.
- NRVO (Named Return Value Optimization): Otimização de valor de retorno nomeado, aplicável quando o objeto retornado tem um nome.

## 6. Exemplo de RVO e NRVO

- Exemplo de RVO:

```
Exemplo criarExemplo() {  
  
    return Exemplo(); // RVO pode ocorrer aqui  
  
}  
  
int main() {  
  
    Exemplo e = criarExemplo(); // Elisão de cópia através de RVO
```

```
    return 0;
}
```

- Exemplo de NRVO:

```
Exemplo criarExemplo() {
    Exemplo e;
    return e; // NRVO pode ocorrer aqui
}
```

```
int main() {
    Exemplo e = criarExemplo(); // Elisão de cópia através de NRVO
    return 0;
}
```

## Benefícios da Elisão de Cópia

- Desempenho: Reduz o overhead de criação e destruição de objetos temporários.
- Eficiência: Melhora a eficiência do código, eliminando operações desnecessárias.
- Segurança: Minimiza o risco de cópias indesejadas e potenciais erros associados.

## Dicas de Boas Práticas

- Clareza de Código: Escreva código de forma que o compilador possa aplicar elisão de cópia facilmente.
- Otimizações do Compilador: Confie nas otimizações do compilador para melhorar o desempenho do código.

- C++17 e Posteriores: Utilize as versões mais recentes do C++ para aproveitar os benefícios das otimizações obrigatórias.

Esta seção abrange os conceitos sobre elisão de cópia em C++. Para mais detalhes, consulte a documentação oficial: [https://en.cppreference.com/w/cpp/language/copy\\_elision](https://en.cppreference.com/w/cpp/language/copy_elision)