



Residência em Software

Residência em Tecnologia da Informação e Comunicação

MÉTODOS ESTÁTICOS

Professores:

Alvaro Degas Coelho,
Edgar Alexander,
Esbel Thomas Valero,
Helder Conceição Almeida

INSTITUIÇÃO EXECUTORA



UESC

COORDENADORA



APOIO

MINISTÉRIO DA
CIÊNCIA, TECNOLOGIA
E INOVAÇÃO



Métodos Estáticos ou Métodos de Classe

- Em C++, os métodos estáticos (também chamados de métodos de classe) são funções que pertencem à classe em vez de pertencerem a instâncias específicas dessa classe.
- Isso significa que eles podem ser chamados sem criar uma instância da classe.
- Métodos estáticos são úteis para realizar operações que não dependem do estado de uma instância específica da classe, mas estão relacionadas à classe como um todo.



Residência
em Software

Sintaxe

```
class MinhaClasse {  
public:  
    // Método estático  
    static void meuMetodoEstatico() {  
        // Implementação do método  
    }  
};  
  
int main() {  
    // Chamando o método estático sem criar uma instância da classe  
    MinhaClasse::meuMetodoEstatico();  
  
    return 0;  
}
```

Características dos Métodos Estáticos

- Eles **não podem** acessar membros de dados não estáticos da classe diretamente, a menos que tenham uma instância da classe para fazê-lo.
- Os métodos estáticos não têm acesso a **this**, pois eles não operam em instâncias específicas da classe.
- Eles podem ser chamados usando o nome da classe seguido pelo operador de resolução de escopo ::.

Características dos Métodos Estáticos

- **Métodos estáticos** são frequentemente usados para **funções utilitárias, métodos de fábrica** e outras operações que não precisam de um objeto específico da classe.
- Eles **não podem** ser declarados como **virtual**, pois não estão relacionados à **herança de objetos de instância**.
- **Variáveis estáticas** dentro de **métodos estáticos** mantêm seu valor entre as chamadas, como variáveis estáticas em funções comuns.
- A utilização de **métodos estáticos** pode **simplificar o código e melhorar o desempenho** em casos em que você precisa realizar operações relacionadas à classe, mas não dependem do estado de instâncias específicas da classe.

Exemplo

```
#include <iostream>

class MinhaClasse {
public:
    static int contador; // Variável estática compartilhada por todas as instâncias
    static void MetodoEstatico() {
        std::cout << "Este é um método estático!" << std::endl;
    }

    void MetodoNaoEstatico() {
        std::cout << "Este é um método não estático!" << std::endl;
    }
};
```



Residência
em Software

Exemplo

```
// Inicializando a variável estática da classe
int MinhaClasse::contador = 0;

int main() {
    // Chamando um método estático
    MinhaClasse::MetodoEstatico();

    // Criando instâncias da classe
    MinhaClasse objeto1;
    MinhaClasse objeto2;

    // Chamando um método não estático em uma instância
    objeto1.MetodoNaoEstatico();

    // Acessando uma variável estática
    std::cout << "Contador: " << MinhaClasse::contador << std::endl;

    return 0;
}
```

Situações onde é recomendável Métodos Estáticos

- **Funções Utilitárias**

- Podem ser usados para agrupar funções utilitárias que não precisam de um estado de instância.
- Por exemplo, **funções de conversão**, **funções matemáticas simples** ou **funções de formatação de texto** podem ser implementadas como métodos estáticos.



```
class Utilitarios {  
public:  
    static double ConverterParaCelsius(double fahrenheit) {  
        return (fahrenheit - 32.0) * 5.0 / 9.0;  
    }  
};  
  
int main() {  
    double tempFahrenheit = 68.0;  
    double tempCelsius = Utilitarios::ConverterParaCelsius(tempFahrenheit);  
    return 0;  
}
```

Situações onde é recomendável Métodos Estáticos

- **Contagem de Instâncias**

- Você pode usar métodos estáticos para rastrear informações sobre todas as instâncias da classe.
- Isso é útil quando você precisa manter um contador global ou fazer algo sempre que uma nova instância da classe for criada.



Residência
em Software

```
class MinhaClasse {  
private:  
    static int contador;  
  
public:  
    MinhaClasse() {  
        contador++;  
    }  
  
    static int GetContador() {  
        return contador;  
    }  
};  
  
int MinhaClasse::contador = 0;  
  
int main() {  
    MinhaClasse objeto1;  
    MinhaClasse objeto2;  
    std::cout << "Número de instâncias: " << MinhaClasse::GetContador() <<  
    return 0;  
}
```

Situações onde é recomendável Métodos Estáticos

- **Funcionalidades Compartilhadas**
 - Métodos estáticos podem ser usados para implementar funcionalidades que são compartilhadas por todas as instâncias da classe, como um **sistema de cache**, **gerenciamento de configurações** ou **funções de log**.

Situações onde é recomendável Métodos Estáticos

- **Facilidade de Uso**

- Às vezes, métodos estáticos simplificam o uso da classe, pois não é necessário criar uma instância da classe para acessar uma função específica. Isso pode tornar o código mais legível e direto.

Considerações...

- É importante lembrar que o uso excessivo de métodos estáticos pode levar a um **design inflexível** e **dificultar a manutenção** e o **teste do código**.
- Portanto, é essencial escolher o uso de métodos estáticos com sabedoria e considerar a estrutura geral do seu programa.
- **Métodos não estáticos ainda são a escolha mais comum** quando você precisa de funcionalidades associadas a instâncias específicas da classe.

Exercícios

Calculadora Estática:

Crie uma classe Calculadora com métodos estáticos para realizar operações matemáticas simples, como adição, subtração, multiplicação e divisão. Peça ao usuário para inserir números e uma operação, e use os métodos estáticos para realizar o cálculo.

Exercícios

Contador Estático:

Crie uma classe Contador que tenha um método estático para contar o número de instâncias da classe criadas. Crie algumas instâncias da classe e imprima o número total de instâncias.

Exercícios

Validador de Senha Estático:

Crie uma classe Senha com um método estático ValidadorSenha que verifique se uma senha atende a critérios específicos (por exemplo, ter pelo menos 8 caracteres, conter letras maiúsculas e minúsculas, números e caracteres especiais). Solicite ao usuário para inserir uma senha e use o método estático para validar.

Exercícios

Gerador de Senha Estático:

Na mesma classe do exercício anterior, crie um método estático que gere senhas aleatórias com base em critérios específicos (os mesmos do exercício anterior). Use o método estático para gerar uma senha e exibi-la.