Classes em C++

Funções de Membros Não Estáticos em C++

Introdução

Funções de membros não estáticos são funções associadas a instâncias de uma classe. Elas podem acessar e modificar os membros de dados não estáticos da instância e são fundamentais para a implementação de comportamentos específicos de objetos em programação orientada a objetos.

1. Definição e Sintaxe

- Definição: Funções de membros não estáticos são funções declaradas dentro da definição de uma classe que operam em instâncias dessa classe.
- Sintaxe:

```
class NomeClasse {
public:
    void nomeFuncao();
};
```

2. Declaração e Definição

- Declaração: Funções de membros não estáticos são declaradas dentro da classe.
- Definição: A definição pode ser feita dentro ou fora da definição da classe.
- Exemplo:

Classes em C++

```
class Pessoa {
public:
    std::string nome;
    int idade;

    void apresentar();  // Declaração
};

void Pessoa::apresentar() {  // Definição fora da classe
    std::cout << "Nome: " << nome << ", Idade: " << idade << std::endl;
}

int main() {
    Pessoa p{"Marcos", 20};
    p.apresentar();
    return 0;
}
```

3. Acesso e Modificação de Membros de Dados

- Definição: Funções de membros não estáticos podem acessar e modificar membros de dados da instância da classe.
- Exemplo:

```
class ContaBancaria {
private:
```

```
    double saldo;

public:
    ContaBancaria(double saldoInicial) : saldo(saldoInicial) {}

    void depositar(double valor) {
        saldo += valor;
    }

    void sacar(double valor) {
        saldo -= valor;
    }

    double obterSaldo() const {
        return saldo;
    }
};

int main() {
    ContaBancaria conta(1000.0);
    conta.depositar(200.0);
    conta.sacar(150.0);
    std::cout << "Saldo: " << conta.obterSaldo() << std::endl;
    return 0;
}
```

Classes em C++

4. Funções Constantes

- Definição: Funções constantes não podem modificar membros de dados da instância da classe.
- Sintaxe:

```
class Exemplo {
public:
    int valor;

    void funcaoConstante() const {
        // valor = 10;  // Erro: não pode modificar membros
    }
};
```

5. Sobrecarga de Funções

- Definição: Funções de membros não estáticos podem ser sobrecarregadas com base na assinatura da função (número e tipo dos parâmetros).
- Exemplo:

```
class Calculadora {
public:
    int somar(int a, int b) {
        return a + b;
    }
```

```
    double somar(double a, double b) {
        return a + b;
    }
};

int main() {
    Calculadora calc;
    std::cout << "Soma de inteiros: " << calc.somar(2, 3) << std::endl;
    std::cout << "Soma de doubles: " << calc.somar(2.5, 3.5) << std::endl;
    return 0;
}
```

6. Funções de Membros Virtuais

- Definição: Funções virtuais permitem que classes derivadas substituam a implementação de uma função de membro da classe base.
- Exemplo:

```
class Forma {
public:
    virtual void desenhar() {
        std::cout << "Desenhar forma genérica" << std::endl;
    }
};

class Circulo : public Forma {
```

```
public:
    void desenhar() override {
        std::cout << "Desenhar círculo" << std::endl;
    }
};

int main() {
    Forma* forma = new Circulo();
    forma->desenhar();  // Chamará Circulo::desenhar
    delete forma;
    return 0;
}
```

7. Funções Inline

- Definição: Funções inline são expandidas no ponto de chamada, reduzindo a sobrecarga de chamadas de função.
- Sintaxe:

```
class Exemplo {
public:
    inline void funcaoInline() {
        std::cout << "Função inline" << std::endl;
    }
};
```

```
int main() {
    Exemplo ex;
    ex.funcaoInline();
    return 0;
}
```

Dicas de Boas Práticas

- Encapsulamento: Use funções de membros para encapsular a lógica que manipula os dados da classe.
- Constância: Declare funções como `const` sempre que possível para garantir que não modifiquem o estado do objeto.
- Virtualidade: Use funções virtuais para permitir o polimorfismo e a substituição em classes derivadas.
- Modularidade: Mantenha as definições de funções curtas e focadas em uma única responsabilidade.

Esta seção abrange os conceitos sobre funções de membros não estáticos em C++. Para mais detalhes, consulte a documentação oficial: https://en.cppreference.com/w/cpp/language/member_functions