

Classes em C++

Declaração de Classe em C++

Introdução

Declarações de classes em C++ são fundamentais para a programação orientada a objetos. Elas definem a estrutura e o comportamento dos objetos, especificando membros de dados (atributos) e membros de funções (métodos) que operam sobre esses dados.

1. Definição e Sintaxe

- Definição: A declaração de uma classe define seu layout na memória e os membros que ela contém.

- Sintaxe:

```
class NomeClasse {  
  
    public:  
  
        // Membros públicos  
  
        tipo1 membro1;  
  
        tipo2 metodo1();  
  
  
    private:  
  
        // Membros privados  
  
        tipo3 membro2;  
  
        tipo4 metodo2();  
  
};
```

Classes em C++

2. Membros Públicos e Privados

- Membros Públicos: Acessíveis de qualquer lugar onde o objeto da classe é visível.
- Membros Privados: Acessíveis apenas dentro da própria classe ou por amigos da classe.
- Exemplo:

```
class Exemplo {  
  
public:  
  
    int publico;  
  
    void metodoPublico() {  
        std::cout << "Método público" << std::endl;  
    }  
  
private:  
  
    int privado;  
  
    void metodoPrivado() {  
        std::cout << "Método privado" << std::endl;  
    }  
};
```

3. Construtores e Destrutores

- Construtor: Inicializa um objeto da classe.

Classes em C++

- Destrutor: Limpa recursos alocados pela classe.

- Exemplo:

```
class Pessoa {  
  
public:  
  
    Pessoa(std::string nome, int idade) : nome(nome), idade(idade) {}  
  
    ~Pessoa() {  
        // Código de limpeza  
    }  
  
    void apresentar() {  
        std::cout << "Nome: " << nome << ", Idade: " << idade << std::endl;  
    }  
  
private:  
  
    std::string nome;  
  
    int idade;  
};
```

4. Métodos Constantes

- Definição: Métodos que não alteram os membros da classe.

- Sintaxe:

```
class Exemplo {  
  
public:
```

Classes em C++

```
void metodoConstante() const {  
    // Não pode alterar membros da classe  
}
```

private:

```
    int valor;  
};
```

5. Herança

- Definição: Permite criar uma nova classe baseada em uma classe existente.
- Sintaxe:

```
class Base {  
  
public:  
    void metodoBase() {}  
};
```

```
class Derivada : public Base {  
  
public:  
    void metodoDerivado() {}  
};
```

6. Polimorfismo

- Definição: Permite que métodos em classes derivadas tenham diferentes implementações.
- Exemplo:

Classes em C++

```
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();  
  
    delete forma;  
  
    return 0;  
  
}
```

7. Classes Aninhadas

- Definição: Classes definidas dentro de outras classes.
- Exemplo:

Classes em C++

```
class Externa {  
  
public:  
  
    class Interna {  
  
public:  
  
        void metodoInterno() {  
  
            std::cout << "Método da classe interna" << std::endl;  
  
        }  
  
};  
  
  
    void metodoExterno() {  
  
        Interna interna;  
  
        interna.metodoInterno();  
  
    }  
  
};
```

8. Declaração de Forward

- Definição: Declaração de uma classe antes de sua definição completa.

- Exemplo:

```
class ClasseA; // Declaração de forward
```

```
class ClasseB {  
  
public:  
  
    void metodo(ClasseA* a);  
  
};
```

Classes em C++

```
class ClasseA {  
  
public:  
  
    void metodo() {}  
  
};  
  
void ClasseB::metodo(ClasseA* a) {  
  
    a->metodo();  
  
}
```

Dicas de Boas Práticas

- Encapsulamento: Mantenha os dados privados e forneça métodos públicos para acesso.
- Clareza: Use nomes de membros e métodos descritivos.
- Modularidade: Divida o código em classes e métodos coesos.
- Documentação: Comente o código para explicar a finalidade e o comportamento dos membros da classe.

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