Classes em C++

Introdução

Classes são um dos principais pilares da programação orientada a objetos (OOP) em C++. Elas permitem agrupar dados e funções que operam sobre esses dados em uma única unidade, facilitando a organização, reutilização e manutenção do código.

1. Definição e Sintaxe

- Definição: Uma classe é uma estrutura de dados que contém membros de dados (variáveis) e membros de funções (métodos).
- Sintaxe:

```
class NomeClasse {
public:
    // Membros públicos
    tipo1 membro1;
    tipo2 metodo1();

private:
    // Membros privados
    tipo3 membro2;
    tipo4 metodo2();
};
```

Classes em C++

2. Declaração e Implementação de Classe

- Declaração: Definição da estrutura da classe, especificando seus membros de dados e funções.
- Implementação: Definição das funções membros fora da classe, usando o operador de resolução de escopo `::`.
- Exemplo:

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

    void apresentar();

private:
    std::string cpf;
};

void Pessoa::apresentar() {
    std::cout << "Nome: " << nome << ", Idade: " << idade << std::endl;
}
```

3. Encapsulamento

- Definição: O encapsulamento é a propriedade de restringir o acesso direto a alguns componentes

da classe, protegendo a integridade dos dados.

- Acessibilidade:
  - `public`: Membros acessíveis de qualquer lugar.
  - `private`: Membros acessíveis apenas dentro da classe.
  - `protected`: Membros acessíveis dentro da classe e em classes derivadas.
- Exemplo:

```
class ContaBancaria {
public:
    void depositar(double valor);
    void sacar(double valor);
    double obterSaldo() const;

private:
    double saldo;
};
```

4. Construtores e Destrutores

- Construtores: Funções especiais que são chamadas quando um objeto da classe é criado. Usados para inicializar membros da classe.
- Destrutores: Funções especiais que são chamadas quando um objeto da classe é destruído. Usados para liberar recursos.
- Sintaxe:

```
class Exemplo {
public:
```

```
    Exemplo();  // Construtor
    ~Exemplo(); // Destrutor
};

Exemplo::Exemplo() {
    // Inicialização
}

Exemplo::~Exemplo() {
    // Limpeza
}
```

5. Herança

- Definição: A herança permite criar uma nova classe (classe derivada) baseada em uma classe existente (classe base), reutilizando e estendendo seu comportamento.
- Sintaxe:

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

class ClasseDerivada : public ClasseBase {
public:
    void metodoDerivado();
```

```
};
```

6. Polimorfismo

- Definição: O polimorfismo permite que uma função ou método tenha comportamentos diferentes com base no objeto que está sendo chamado.
- 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. Classes e Métodos Constantes

- Classe Constante: Objetos constantes não podem modificar seus membros.
- Métodos Constantes: Métodos que não modificam os membros da classe.
- Sintaxe:

```
class Exemplo {
public:
    void metodo() const;
private:
    int valor;
};

void Exemplo::metodo() const {
    // Não pode modificar membros
}
```

Dicas de Boas Práticas

- Encapsulamento: Sempre que possível, mantenha os dados privados e forneça métodos públicos para acessar e modificar esses dados.
- Construtores: Inicialize todos os membros no construtor.
- Destrutores: Libere todos os recursos no destrutor para evitar vazamentos de memória.

Classes em C++

- Herança: Use herança quando houver uma relação claro de "é um" entre a classe base e a classe derivada.
- Polimorfismo: Use polimorfismo para permitir que métodos e funções lidem com objetos de diferentes classes de maneira uniforme.

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