

Classes em C++

Construtor Padrão em C++

Introdução

O construtor padrão em C++ é um construtor que pode ser chamado sem nenhum argumento. Ele é usado para inicializar objetos quando nenhuma informação específica é fornecida. Se uma classe não define explicitamente nenhum construtor, o compilador gera automaticamente um construtor padrão.

1. Definição e Sintaxe

- Definição: Um construtor padrão é um construtor que pode ser invocado sem fornecer argumentos.

- Sintaxe:

```
class NomeClasse {  
  
public:  
  
    NomeClasse(); // Declaração do construtor padrão  
  
};
```

2. Construtor Padrão Implicitamente Definido

- Definição: Se nenhuma forma de construtor for fornecida pela classe, o compilador gera automaticamente um construtor padrão.

- Exemplo:

Classes em C++

```
class Exemplo {  
  
public:  
  
    int valor;  
  
};  
  
int main() {  
  
    Exemplo obj; // Chama o construtor padrão implicitamente definido  
  
    return 0;  
  
}
```

3. Construtor Padrão Explicitamente Definido

- Definição: Um construtor padrão pode ser explicitamente definido pelo programador.
- Exemplo:

```
class Exemplo {  
  
public:  
  
    int valor;  
  
  
    Exemplo() { // Construtor padrão explicitamente definido  
  
        valor = 0;  
  
    }  
  
};  
  
int main() {  
  
    Exemplo obj; // Chama o construtor padrão explicitamente definido
```

Classes em C++

```
    return 0;  
}
```

4. Construtor Padrão e Inicialização de Membros

- Definição: Membros da classe podem ser inicializados dentro do corpo do construtor padrão ou usando uma lista de inicializadores.

- Exemplo:

```
class Exemplo {  
  
private:  
  
    int valor1;  
  
    int valor2;  
  
  
public:  
  
    Exemplo() : valor1(0), valor2(0) { // Inicialização com lista de inicializadores  
        // Código adicional  
    }  
};  
  
  
int main() {  
  
    Exemplo obj;  
  
    return 0;  
}
```

5. Construtores Padrão e Classes Derivadas

Classes em C++

- Definição: Quando uma classe derivada é instanciada, o construtor padrão da classe base é chamado automaticamente se não houver inicializadores explícitos.

- Exemplo:

```
class Base {  
  
public:  
  
    Base() {  
  
        std::cout << "Construtor da classe base" << std::endl;  
  
    }  
  
};  
  
class Derivada : public Base {  
  
public:  
  
    Derivada() {  
  
        std::cout << "Construtor da classe derivada" << std::endl;  
  
    }  
  
};  
  
int main() {  
  
    Derivada obj;  
  
    return 0;  
  
}
```

6. Construtor Padrão `default`

Classes em C++

- Definição: Um construtor padrão pode ser explicitamente declarado como `default` para indicar que o compilador deve gerar a implementação padrão.

- Exemplo:

```
class Exemplo {  
  
public:  
  
    Exemplo() = default; // Solicita ao compilador que gere o construtor padrão  
  
};  
  
int main() {  
  
    Exemplo obj;  
  
    return 0;  
  
}
```

7. Desabilitando o Construtor Padrão

- Definição: O construtor padrão pode ser desabilitado explicitamente para impedir a criação de objetos sem argumentos.

- Exemplo:

```
class Exemplo {  
  
public:  
  
    Exemplo() = delete; // Desabilita o construtor padrão  
  
    Exemplo(int valor) {  
  
        // Construtor parametrizado  
  
    }  
  
};
```

Classes em C++

```
int main() {  
    // Exemplo obj; // Erro: construtor padrão está desabilitado  
  
    Exemplo obj(10); // Correto  
  
    return 0;  
}
```

Dicas de Boas Práticas

- Inicialização: Sempre inicialize os membros da classe no construtor padrão para garantir que os objetos estejam em um estado válido.
- Uso de `default`: Use `default` para indicar claramente que a implementação padrão do construtor deve ser gerada pelo compilador.
- Desabilitação: Desabilite o construtor padrão se a criação de objetos sem argumentos não fizer sentido para a lógica da classe.

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