

Classes em C++

Especificador de Função Virtual em C++

Introdução

O especificador de função virtual em C++ é usado para permitir que funções membros de uma classe base sejam substituídas por funções membros de uma classe derivada. Isso é fundamental para o polimorfismo, permitindo que diferentes tipos de objetos sejam tratados de maneira uniforme através de ponteiros ou referências à classe base.

1. Definição e Sintaxe

- Definição: Uma função virtual é uma função membro de uma classe que pode ser substituída em uma classe derivada.

- Sintaxe:

```
class Base {  
  
public:  
  
    virtual void funcaoVirtual();  
  
};
```

2. Exemplo de Função Virtual

- Exemplo: Uma função virtual permite que uma função membro da classe derivada seja chamada, mesmo quando estamos lidando com um ponteiro ou referência à classe base.

```
#include <iostream>
```

Classes em C++

```
class Base {  
  
public:  
  
    virtual void mostrar() {  
  
        std::cout << "Classe Base" << std::endl;  
  
    }  
  
};
```

```
class Derivada : public Base {  
  
public:  
  
    void mostrar() override {  
  
        std::cout << "Classe Derivada" << std::endl;  
  
    }  
  
};
```

```
int main() {  
  
    Base* b = new Derivada();  
  
    b->mostrar(); // Chamará Derivada::mostrar  
  
    delete b;  
  
    return 0;  
  
}
```

3. Polimorfismo

- Definição: O polimorfismo permite que objetos de diferentes classes derivadas sejam tratados

Classes em C++

como objetos da classe base. A função correta é chamada com base no tipo do objeto real.

- Exemplo:

```
class Forma {  
  
public:  
  
    virtual void desenhar() {  
  
        std::cout << "Desenhar forma" << std::endl;  
  
    }  
  
};  
  
class Circulo : public Forma {  
  
public:  
  
    void desenhar() override {  
  
        std::cout << "Desenhar círculo" << std::endl;  
  
    }  
  
};  
  
class Quadrado : public Forma {  
  
public:  
  
    void desenhar() override {  
  
        std::cout << "Desenhar quadrado" << std::endl;  
  
    }  
  
};  
  
int main() {  
  
    Forma* formas[] = {new Circulo(), new Quadrado()};
```

Classes em C++

```
for (Forma* forma : formas) {  
    forma->desenhar();  
}  
  
for (Forma* forma : formas) {  
    delete forma;  
}  
  
return 0;  
}
```

4. Funções Virtuais Puras

- Definição: Uma função virtual pura é uma função virtual que não tem implementação na classe base. Classes que contêm funções virtuais puras são classes abstratas e não podem ser instanciadas.

- Sintaxe:

```
class Base {  
  
public:  
  
    virtual void funcaoVirtualPura() = 0;  
  
};
```

- Exemplo:

```
class Forma {  
  
public:  
  
    virtual void desenhar() = 0; // Função virtual pura  
  
};
```

Classes em C++

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

```
int main() {  
  
    Forma* f = new Circulo();  
  
    f->desenhar();  
  
    delete f;  
  
    return 0;  
  
}
```

5. Destrutores Virtuais

- Definição: Se uma classe tem funções virtuais, é uma boa prática declarar o destrutor como virtual para garantir que o destrutor correto seja chamado para objetos de classes derivadas.

- Exemplo:

```
class Base {  
  
public:  
  
    virtual ~Base() {  
  
        std::cout << "Destrutor de Base" << std::endl;  
  
    }  
  
};
```

Classes em C++

```
};

class Derivada : public Base {
public:
    ~Derivada() {
        std::cout << "Destrutor de Derivada" << std::endl;
    }
};

int main() {
    Base* b = new Derivada();
    delete b; // Chamará ~Derivada e depois ~Base
    return 0;
}
```

Dicas de Boas Práticas

- Virtualidade: Use funções virtuais para permitir substituições em classes derivadas.
- Destrutores Virtuais: Sempre declare destrutores como virtuais em classes base que possuem outras funções virtuais.
- Funções Virtuais Puras: Use funções virtuais puras para definir interfaces em classes base abstratas.

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