



Residência
em Software

Classes Abstratas

Professores:

Álvaro Coelho, Edgar Alexander, Esbel
Valero e Hélder Almeida

INSTITUIÇÃO EXECUTORA



UESC

COORDENADORA



APOIO

MINISTÉRIO DA
CIÊNCIA, TECNOLOGIA
E INOVAÇÃO



Classes Abstratas

“As classes abstratas agem como expressões de conceitos gerais das quais classes mais específicas podem ser derivadas.”

learn.microsoft.com

Classes Abstratas

- Uma classe abstrata em C++ é uma classe que não pode ser instanciada diretamente.
- Em vez disso, ela serve como uma base para outras classes derivadas.
- Uma classe abstrata pode conter métodos puramente virtuais, que são declarados, mas não implementados na classe base.
- Estes métodos devem ser implementados pelas classes derivadas.

Classes Abstratas

- Você cria uma classe abstrata declarando pelo menos uma **função membro virtual pura**.
- Essa é uma **função virtual declarada** usando a sintaxe de **especificador pure** (`= 0`).
- Classes derivadas da classe abstrata **devem implementar** a função virtual pura, ou **serão também classes abstratas**.

Classes Abstratas

```
class Forma {  
public:  
    virtual void desenhar() const = 0; // Método puro virtual  
    virtual double calcularArea() const = 0;  
    // ... outros métodos comuns  
};
```

Neste exemplo, Forma é uma classe abstrata que define uma interface comum para diferentes formas.

As classes derivadas, como Circulo e Retangulo, devem implementar obrigatoriamente os métodos desenhar() e calcularArea().

Restrições sobre classes abstratas

As classes abstratas **não podem** ser usadas para:

- Variáveis ou dados de membro
- Tipos de argumento
- Tipos de retorno de função
- Tipos de conversões explícitas

Outra restrição é que se o construtor para uma classe abstrata chamar uma função virtual pura, direta ou indiretamente, o resultado será indefinido. No entanto, os construtores e os destruidores para classes abstratas podem chamar outras funções de membro.

Funções Virtuais

- **Funções virtuais** são métodos declarados na classe base que podem ser sobrescritos (ou redefinidos) por classes derivadas.
- A **palavra-chave virtual** é usada na declaração da função na classe base.
- Uma **função virtual** em uma classe base **tem uma implementação padrão**, mas essa implementação pode ser substituída por uma implementação específica nas classes derivadas.

Funções Virtuais Puras

- **Funções virtuais puras** são funções virtuais que são declaradas na classe base, mas não fornecem uma implementação padrão.
- A função virtual pura é declarada com a adição do "**= 0**" ao final da declaração, **indicando que não há implementação na classe base** e que as classes derivadas devem fornecer suas próprias implementações.

Funções virtuais puras definidas

Funções virtuais puras em classes abstratas podem ser definidas ou ter uma implementação.

Você só pode chamar essas funções usando a sintaxe totalmente qualificada:

abstract-class-name::function-name()



Funções virtuais puras definidas

```
class base
{
public:
    base() {}
    // To define the virtual destructor outside the class:
    virtual ~base() = 0;
    // Microsoft-specific extension to define it inline:
    // virtual ~base() = 0 {};
};

base::~~base() {} // required if not using Microsoft extension
```



Funções virtuais puras definidas

```
class derived : public base
{
public:
    derived() {}
    ~derived() {}
};

int main()
{
    derived aDerived; // destructor called when it goes out of scope
}
```