

Revisão de POO - NOM2

Abstração

Extrair do mundo apenas o que é relevante ao sistema

Encapsulamento

Esconder os atributos de outros objetos e a implementação dos métodos de outros programadores

FUNÇÕES FRIENDS
Função definida fora do escopo da classe para acessar os membros públicos e privados dessa classe
(quebra de encapsulamento)

Destrutor

Função-membro especial utilizada para desalocar a memória usada para criação de determinados objetos.

Essa função é chamada imediatamente antes que objeto seja destruído justamente para limpar a memória desse objeto antes que ele seja permanentemente excluído

Métodos de acesso

Métodos utilitários

métodos public

métodos private

Sintaxe de Inicialização

A sintaxe de inicialização de membros é colocada logo após o nome do construtor, pode ser usada sempre.

```
// Sintaxe de inicialização de membros
Pessoa(Pessoa& nome, int idade) : nome(nome), idade(idade) {
    // Corpo do construtor (não necessário para inicialização dos membros)
}
```

Eficiência: Inicializa os membros diretamente, sem precisar de atribuições no corpo do construtor.

Necessário para const e reference: Esses membros só podem ser inicializados dessa forma, já que não podem ser atribuídos após a criação do objeto.

Retornar referência p/ membro private

Retornar uma referência para membros privados pode ser perigoso porque permite que o código externo modifique diretamente os dados da classe, quebrando o encapsulamento e violando a integridade da classe.

Construtor

Mesmo nome da classe, sem retorno, pode ser sobrecarregado

Função membro especial que é invocada na definição do objeto para inicializá-lo.

Valores default: São valores que você define para um ou mais parâmetros de uma função, de modo que, caso o usuário não forneça um valor ao chamar a função, o valor padrão será utilizado (feito na .h)

Interface

São os métodos públicos da classe, o que ela oferece ao mundo.

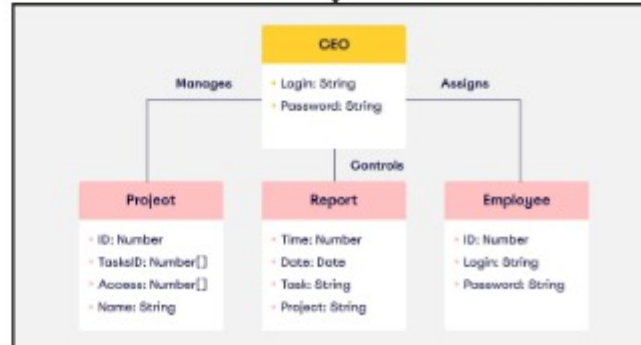
Implementação

Como a classe realiza as operações e funcionalidades definidas pela interface

Interface fica no .hpp (ou .h)
Implementação fica no .cpp

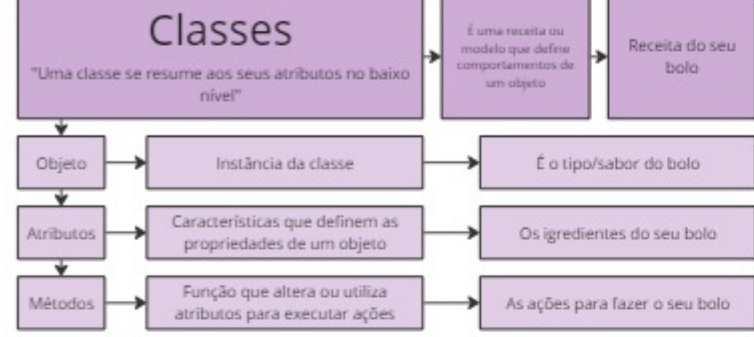
****Modularidade e Organização****
****Flexibilidade e Reuso****
****Facilidade na Manutenção****

UML (Unified Modeling Language)

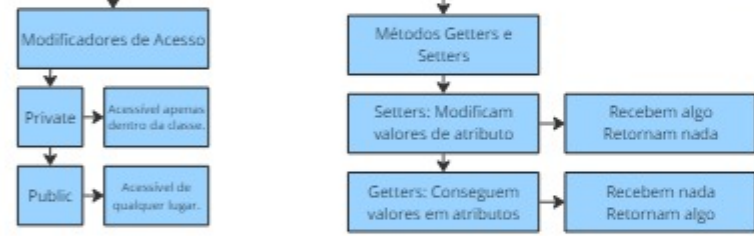


- (sinal de menos): atributos/métodos privados
+ (sinal de mais): atributos/métodos públicos

Classes



C++ 20 em POO



Const

Métodos que não alteram o estado do objeto devem ser definidos como const

Objetos que não precisam/podem ser modificados, e, portanto, devem ser definidos como objetos const

Objeto	Método	É ok?
não-const	não-const	Proibido
não-const	const	Proibido
const	não-const	Não possível
const	const	Proibido

Ponteiro This

Argumento implícito presente em cada função-membro não static da classe

Contém o endereço do objeto a partir do qual foi invocado o método.

Distinguir o parâmetro do atributo quando estes possuírem o mesmo nome

new and delete

Permitem alocar e desalocar memória dinamicamente (em tempo de execução)

new[] delete[] -> Mesma coisa só que para arrays de objetos

Membros Statics

Propriedade da Classe compartilhada por todos os objetos dessa Classe

Sempre existe independente de objetos

Associação

Quando dois objetos se relacionam de alguma forma.

Associação entre classes:
É um tipo de relação que indica que uma classe está conectada a outra de alguma forma, permitindo que objetos de uma classe interajam com objetos da outra.

Composição

Uma Classe tem objetos de outra Classe como membros

Na composição, a parte faz parte do todo e o todo não existe sem a parte.

Agregação

A parte e o todo são independentes. O todo ainda existe mesmo sem a parte, e a parte ainda existe sem o todo