



Programação Orientada a Objeto 2022

Prof. Dr. José Erinaldo da Fonsêca

São Paulo - SP, Março de 2022.

Introdução

- ▶ **Objetivo:** Apresentar o conceito de Encapsulamento.
- ▶ **Pré-Requisitos:** Objeto, Classe, Atributos, Métodos, Métodos Especiais (Get, Set e Construtor).
- ▶ **Tópicos abordados:** Conceitos sobre encapsulamento e código encapsulado.

Pilares de POO



- ▶ Encapsulamento
- ▶ Herança
- ▶ Polimorfismo

Encapsulamento

- ▶ A pilha é uma capsula.
- ▶ Qual o objetivo da capsula?
- ▶ Proteção contra o componente químico. Ex: Zn.
- ▶ Proteção para a pilha de fato funcionar.



Encapsulamento

- ▶ Uma forma de criar um padrão.
- ▶ Por exemplo, das pilhas apresentadas ao lado, a quadrada não serve para o mesmo aparelho que serve as pilhas palitos.
- ▶ Veja que existe um formato, uma intensidade...existem um padrão.



Encapsulamento

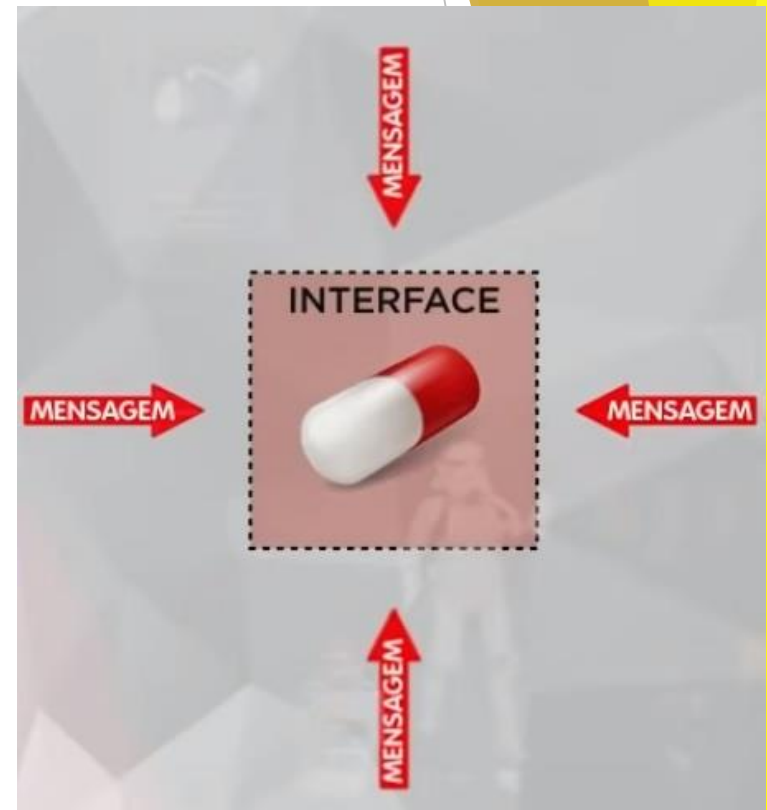
- ▶ Um software encapsulado tem o mesmo padrão.
- ▶ Ele protege o código do usuário.
- ▶ E protege o usuário do código.

Encapsulamento

- ▶ Portanto, encapsular é ocultar partes independentes da implementação, permitindo construir partes invisíveis ao mundo exterior.

Encapsulamento

- ▶ Assim como a pilha tem dois pinos que se conecta com o mundo exterior, o código também tem um contato com o mundo exterior. Chamamos isso de interface.
- ▶ A interface é uma lista de serviços fornecidos por um componente. É o contato com o mundo exterior, que define o que pode ser feito com um objeto dessa classe.



Encapsulamento

Importante!!

- ▶ Encapsular não é obrigatório, mas é uma boa prática para produzir Classes mais eficientes.



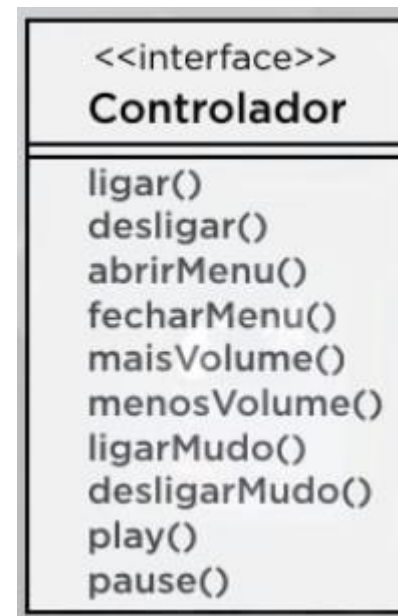
Encapsulamento

Vantagens em Encapsular:

- ▶ Tornar mudanças invisíveis.
- ▶ Facilitar reutilização do código.
- ▶ Reduzir efeitos colaterais.

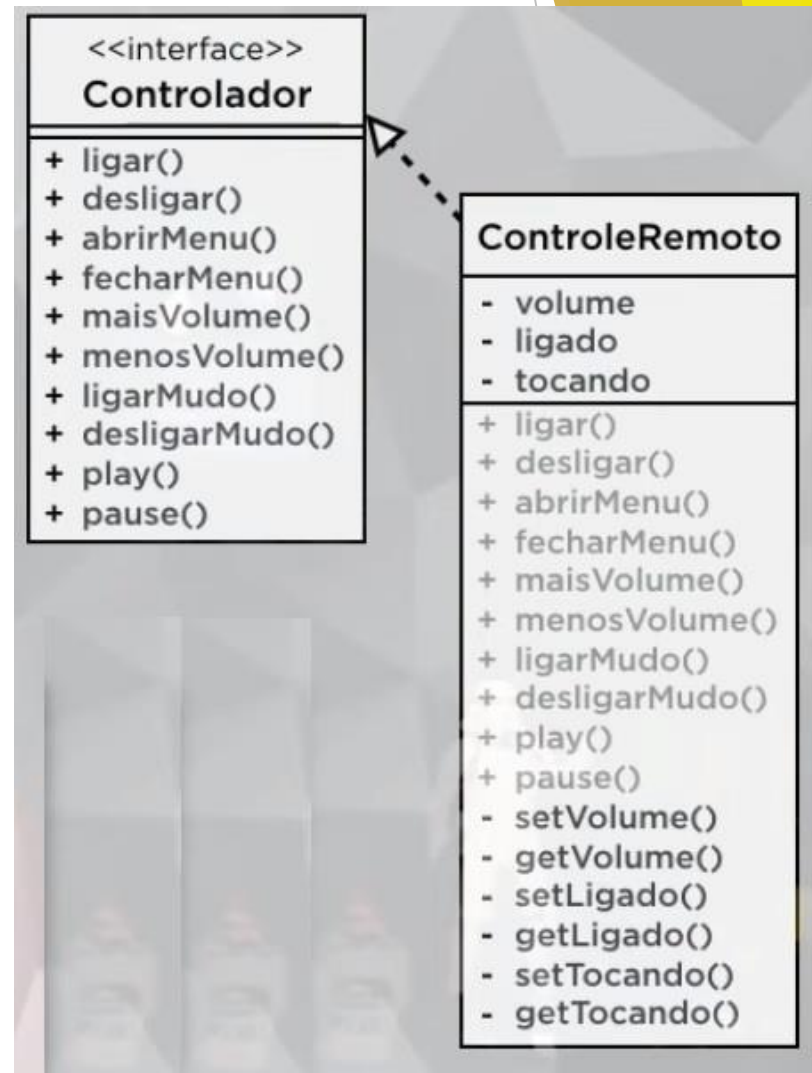
Como encapsular?

- ▶ Uma interface só tem métodos.
- ▶ Estes métodos são chamados de métodos abstratos.
- ▶ Eles não são desenvolvidos na interface, ou seja, não é escrito o código exato do seu funcionamento.
- ▶ Só é dito na interface que tais métodos irão existir.
- ▶ Todos os métodos na interface são Públicos



Como encapsular?

- ▶ Criamos a Classe “Controle Remoto”.
- ▶ Nesta Classe implementamos os códigos de cada método do “Controlador”.
- ▶ Para este exemplo vamos deixar os atributos e os métodos Gets e Sets privado.



Como encapsular?

- ▶ Conforme o modelo do controlador, podemos criar a interface.
- ▶ Veja que temos o termo abstrato para indicar exatamente que nele não será implementado o código.

```
interface Controlador
// Métodos Abstratos
publico abstrato Metodo ligar()
publico abstrato Metodo desligar()
publico abstrato Metodo abrirMenu()
publico abstrato Metodo fecharMenu()
publico abstrato Metodo maisVolume()
publico abstrato Metodo menosVolume()
publico abstrato Metodo ligarMudo()
publico abstrato Metodo desligarMudo()
publico abstrato Metodo play()
publico abstrato Metodo pause()
FimInterface
```

<<interface>>
Controlador

- + ligar()
- + desligar()
- + abrirMenu()
- + fecharMenu()
- + maisVolume()
- + menosVolume()
- + ligarMudo()
- + desligarMudo()
- + play()
- + pause()

Como encapsular?

- ▶ Agora construímos a Classe conforme já fizemos em exemplos anteriores.
- ▶ O que falta agora?



Como encapsular?

- ▶ Falta fazer a ligação.
- ▶ O controle remoto implementa o controlador.
- ▶ Em seguida sobrescrevemos os métodos dentro da classe.



- Implementação do código ligar e desligar.
- Implementação do código AbrirMenu e FecharMenu.



```

publico Metodo abrirMenu()
    Escreva(getLigado())
    Escreva(getVolume())
    Para i = 0 ate getVolume() passo 10 faca
        Escreva("|")
    FimPara
    Escreva(getTocando())
FimMetodo
publico Metodo fecharMenu()
    Escreva("Fechando Menu...")
FimMetodo
  
```



```

publico Metodo maisVolume()
    Se (getLigado()) entao
        setVolume(getVolume() + 1)
    FimSe
FimMetodo
publico Metodo menosVolume()
    Se (getLigado()) entao
        setVolume(getVolume() - 1)
    FimSe
FimMetodo
publico Metodo ligarMudo()
    Se (getLigado() e getVolume() > 0) entao
        setVolume(0)
    FimSe
FimMetodo
publico Metodo desligarMudo()
    Se (getLigado() e getVolume() = 0) entao
        setVolume(50)
    FimSe
FimMetodo
FimClasse

```

- Implementação dos demais métodos.

```

publico Metodo play()
    Se (getLigado() e nao getTocando()) entao
        setTocando(verdadeiro)
    FimSe
FimMetodo
publico Metodo pause()
    Se (getLigado() e getTocando()) entao
        setTocando(falso)
    FimSe
FimMetodo
FimClasse

```

Atividade

- ▶ Desenvolva o código do exemplo dado na aula.

Referências

- ▶ DEITEL, P.; DEITEL, H.; **Java: como programar**. 10ª ed. São Paulo: Pearson Education do Brasil, 2017. E-book
- ▶ SCHILDT, H.; **Java para iniciantes: crie, compile e execute programas Java rapidamente** [recurso eletrônico]. 6. ed. Porto Alegre: Bookman, 2015. E-book
- ▶ SEBESTA, R.W; **Conceitos de linguagem de programação** [recurso eletrônico]. 11ª ed. Porto Alegre: Bookman, 2018. E-book.
- ▶ Acessado em 10/02/2022. <https://www.cursoemvideo.com/curso/java-poo/>



► FIM!!