# Curso POO Teoria #06a - Pilares da POO: Encapsulamento

Pilares da POO

A E H P

Abstração

Encapsulamento

Herança

Polimorfismo

Encapsulamento: pense em uma coisa que tenha cápsula como uma pilha.

O software encapsulado protege o usuário do código e o código do usuário. O software encapsulado tem o mesmo padrão. O software encapsulado não deixa o usuário fazer besteira.

Encapsular: Ocultar partes independentes da implementação permitindo construir partes invisíveis ao mundo exterior.

Um bom objeto encapsulado possui uma interface bem definida. A pilha com seus dois polos que seria sua interface.

VANTAGENS DE SE PROGRAMAR POO COM ENCAPSULAMENTO:

Tornar mudanças invisíveis. Ex : uma pilha trocada não vai ser percebido se a pilha foi trocada por outra. O importante é que ela só vai estar carregando o controle remoto.

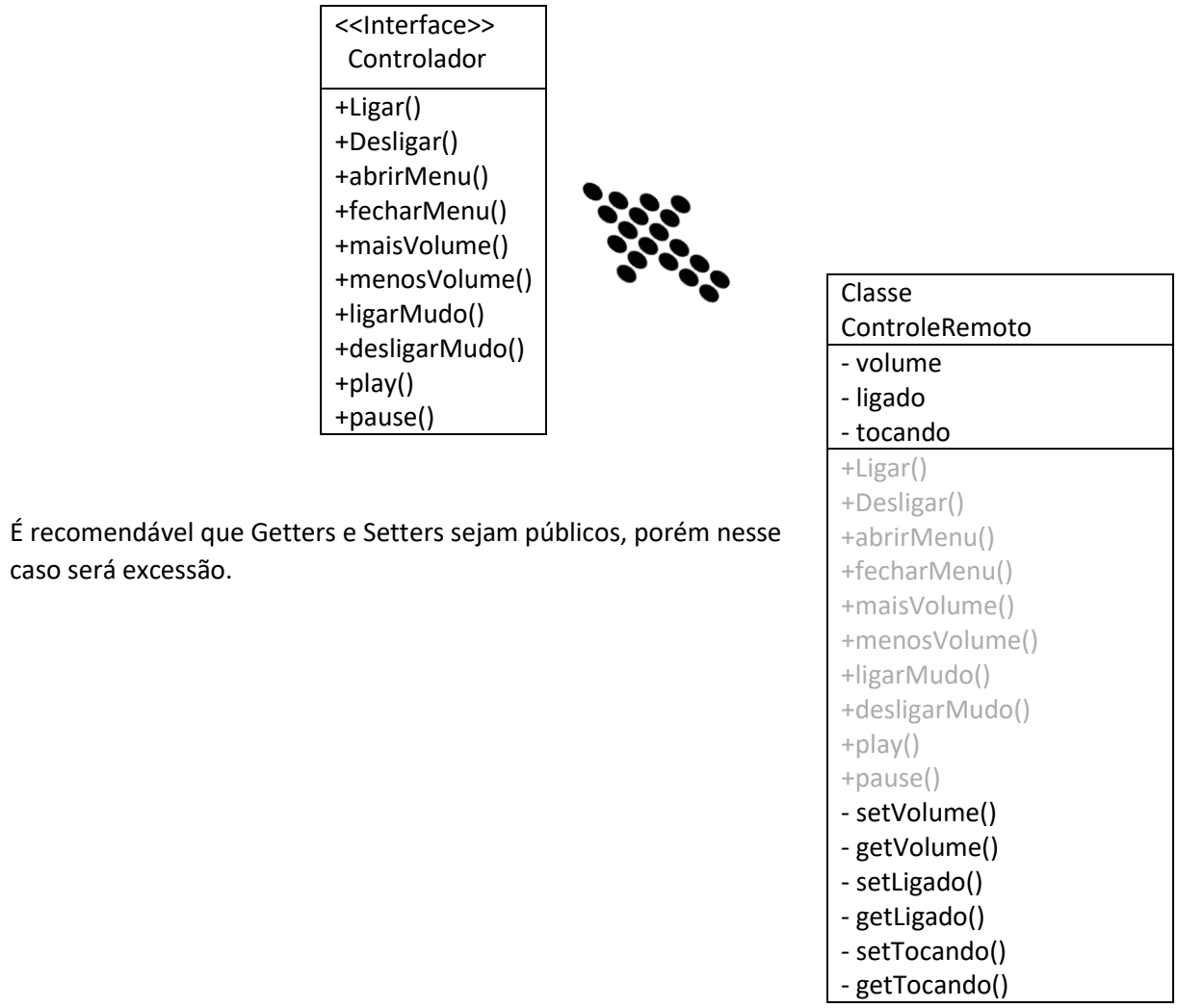
Facilitar reutilização do código. Modelos de encapsulamento fazem a diferença.

Reduzir efeitos colaterais ou seja efeitos danosos

Outro exemplo é quando se coloca uma capa protetora em um controle para proteger a parte interna com botões que pode – se apertar, ou seja o usuário terá interação com a interface o protegendo da parte interna do controle.

Método abstrato: previsto porém não implementado.

Todos os métodos vão ser públicos




É recomendável que Getters e Setters sejam públicos, porém nesse caso será excessão.

Interface Controlador

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 play()

publico abstrato Metodo pause()

FimInterface

```
 classe ControleRemoto
//Atributos
privado inteiro : volume
privado lógico: ligado
privado lógico: tocando
//Métodos Especiais
 publico Metodo Constructor()
volume = 100
ligado = falso
tocando = falso
FimMetodo
 privado Metodo getVolume()
retorne volume
FimMetodo
privado Metodo getLigado()
 retorne volume
FimMetodo
Privado Metodo getTocando()
retorne volume
FimMetodo
privado Metodo setVolume(v: inteiro)
 volume= v
FimMetodo
privado Metodo setLigado(l:logico)
 ligado = l
FimMetodo
privado Metodo setTocando(t: logico)
tocando = t
FimMetodo
FimClasse
```

```
Classe ControleRemoto
 Publico Metodo Ligar()

setLigado(Verdadeiro)
FimMetodo

Publico Metodo Desligar()

setLigado(Falso)
FimMetodo

Publico Metodo abrirMenu()
Escreva(getLigado())
Escreva(getVolume())
Para i =0 ate getVolume() passo 10 faca
Escreva (“  I  ”)
FimPara
Escreva (getTocando())
FimMetodo

Publico Metodo fecharMenu()
Escreva (“ Fechando Menu ”)
FimMetodo

Publico Metodo maisVolume()
Se (getLigado()) então
setVolume(getVolume() + 1)
Fimse
FimMetodo
Publico Metodo menosVolume()
```

```
Se (getLigado()) então
setLigado(getLigado() - 1)
Fimse
FimMetodo

Publico Metodo ligarMudo()
Se (getLigado() e getVolume() > 0) então
setVolume(0)
Fimse
FimMetodo
Publico Metodo desligarMudo()
Se (getLigado() e getVolume() = 0) então
setVolume(100)
Fimse
FimMetodo

Publico Metodo play()
Se (getLigado() e nao getTocando()) então
setTocando(verdadeiro)
Fimse
FimMetodo

Publico Metodo pause()
Se (getLigado() e getTocando()) então
setTocando(falso)
Fimse
FimMetodo
FimClasse
```