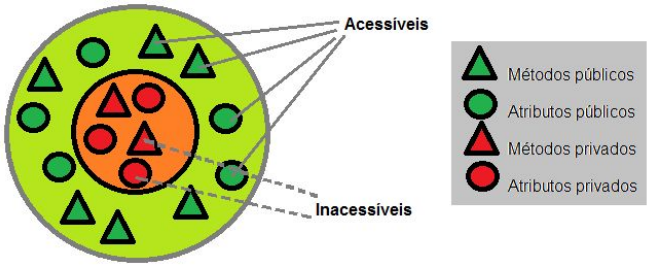
Orientação à objetos

Pilares da POO: Encapsulamento

Aprofundando em POO

- É comum e recomendável que os campos sejam *private*.
  - Isso significa que eles somente podem ser acessados diretamente pela classe onde foram definidos.
  - O acesso aos campos e valores seja feito indiretamente por meio de métodos públicos: get e set.



Michelle Karissa Luciano Carvalho

Orientação à objetos

Pilares da POO: Encapsulamento

- É possível ter a troca de informações entre o objeto e o mundo externo com o envio de mensagem.
- O canal de comunicação para a troca de mensagens em POO é chamado de **interface**.
- Para que a troca de mensagens seja eficaz, a **interface** deve ser bem definida.

Michelle Karissa Luciano Carvalho

Orientação à objetos

Pilares da POO: Encapsulamento

- **Interface**:
  - 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.
    - Um software encapsulado é uma cápsula que se comunica com o meio externo através da interface → fornece aumento de qualidade do projeto.

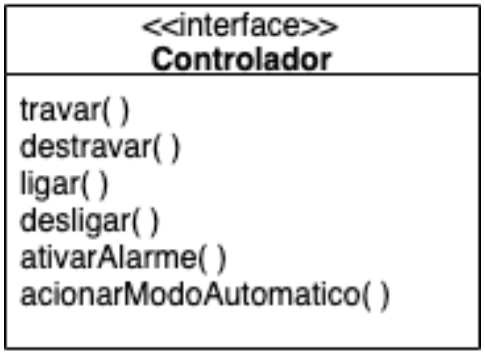
Michelle Karissa Luciano Carvalho

Orientação à objetos

Pilares da POO: Encapsulamento

Exemplo

Objetos



- Criar uma interface para permitir interação com o objeto sem precisar compreender como é o funcionamento interno.
- Interface não tem atributos, apenas métodos.
  - Esses métodos são chamados de abstratos, pois não serão desenvolvidos na interface.
  - Serão apenas declarados para saber a reação.
  - São previstos, mas não implementados.

Michelle Karissa Luciano Carvalho

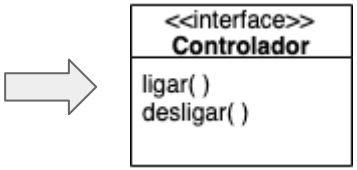
UFBA Universidade Federal da Bahia computação UFBA

Orientação à objetos

Pilares da POO: Encapsulamento

Exemplo

Objetos



A ação de desligar é a mesma (interface), mas a forma de funcionamento é diferente.

- Criar uma interface para permitir interação com o objeto sem precisar compreender como é o funcionamento interno.
- Interface não tem atributos, apenas métodos.
  - Esses métodos são chamados de abstratos, pois não serão desenvolvidos na interface.
  - Serão apenas declarados para saber a reação.
  - São previstos, mas não implementados.

Michelle Karissa Luciano Carvalho

UFBA Universidade Federal da Bahia computação UFBA

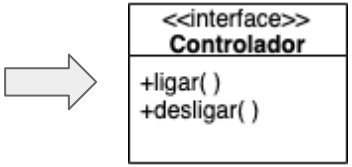
Orientação à objetos

Pilares da POO: Encapsulamento

Exemplo

Objetos

Métodos com visibilidade pública.



A ação de desligar é a mesma (interface), mas a forma de funcionamento é diferente.

- Criar uma interface para permitir interação com o objeto sem precisar compreender como é o funcionamento interno.
- Interface não tem atributos, apenas métodos.
  - Esses métodos são chamados de abstratos, pois não serão desenvolvidos na interface.
  - Serão apenas declarados para saber a reação.
  - São previstos, mas não implementados.

Michelle Karissa Luciano Carvalho

UFBA Universidade Federal da Bahia computação UFBA

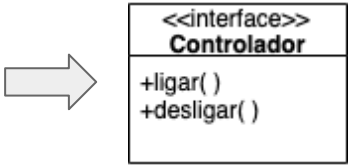
Orientação à objetos

Pilares da POO: Encapsulamento

Exemplo

Objetos

Métodos com visibilidade pública.



A ação de desligar é a mesma (interface), mas a forma de funcionamento é diferente.

- Criar uma interface para permitir interação com o objeto sem precisar compreender como é o funcionamento interno.
- Interface não tem atributos, apenas métodos.
  - Esses métodos são chamados de abstratos, pois não serão desenvolvidos na interface.
  - Serão apenas declarados para saber a reação.
  - São previstos, mas não implementados.

Michelle Karissa Luciano Carvalho

UFBA Universidade Federal da Bahia computação UFBA

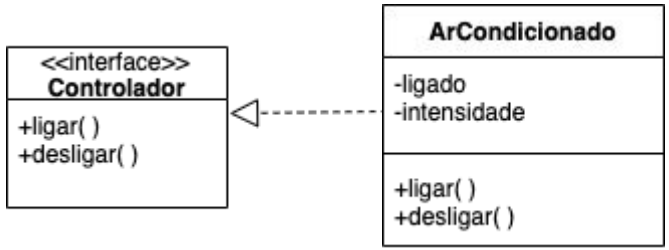
Orientação à objetos

Pilares da POO: Encapsulamento

Exemplo

Métodos com visibilidade pública.

Atributos com visibilidade privada.



- Criar uma interface para permitir interação com o objeto sem precisar compreender como é o funcionamento interno.
- Criar classe (molde) que vai implementar a interface.
  - Essa classe terá como métodos os que foram declarados na interface.

Michelle Karissa Luciano Carvalho

UFBA Universidade Federal da Bahia computação UFBA

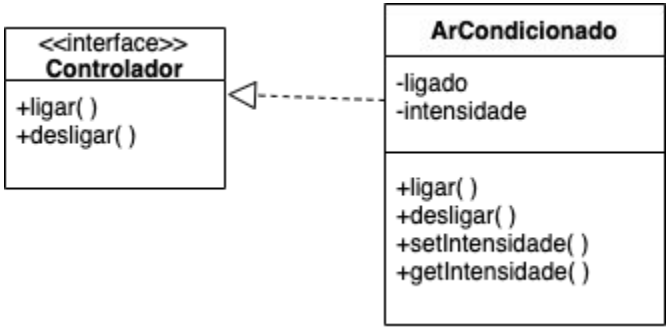
Orientação à objetos

Pilares da POO: Encapsulamento

Exemplo

Métodos com visibilidade pública.

Atributos com visibilidade privada.



- Criar uma interface para permitir interação com o objeto sem precisar compreender como é o funcionamento interno.
- Criar classe (molde) que vai implementar a interface.
  - Essa classe terá como métodos os que foram declarados na interface.
  - Além disso, incluir os métodos adicionais relacionados com os atributos definidos.
    - Como os atributos têm visibilidade "private", devemos definir os métodos get e set para cada um deles.

Michelle Karissa Luciano Carvalho

UFBA Universidade Federal da Bahia computação UFBA

Orientação à objetos

Pilares da POO: Encapsulamento

Exemplo


```
interface Controlador
        //métodos Abstratos
        publico abstrato Metodo ligar( )
        publico abstrato Metodo desligar( )
FimInterface
```

- Interface tem visibilidade pública, pois é acessada por pelas classes definidas.
- Os métodos são abstratos, pois não têm instruções implementadas.

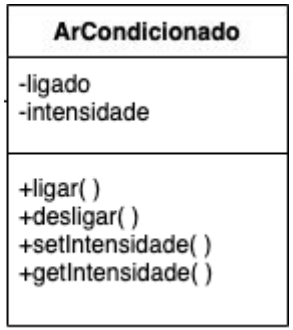
Michelle Karissa Luciano Carvalho

UFBA Universidade Federal da Bahia computação UFBA

Orientação à objetos

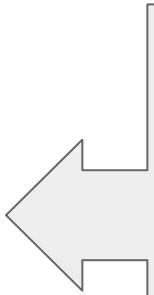
Pilares da POO: Encapsulamento

Exemplo



classe ArCondicionado
(...)

FimClasse



- Interface tem visibilidade pública, pois é acessada por pelas classes definidas.
- Os métodos são abstratos, pois não têm instruções implementadas.

Michelle Karissa Luciano Carvalho

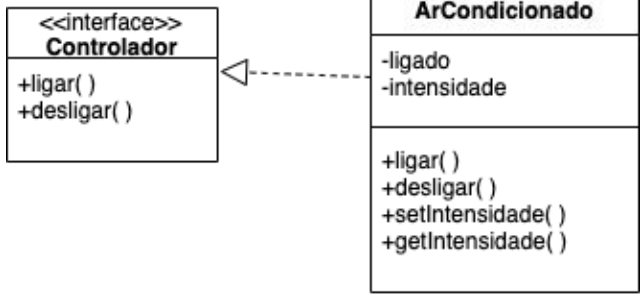
UFBA Universidade Federal da Bahia computação UFBA

```
classe ArCondicionado
        //atributos
        privado logico ligado
        privado inteiro intensidade

        //métodos especiais
        publico Metodo Construtor()
                intensidade = 22
                ligado = falso
        FimMetodo
        privado Metodo getLigado()
                retorne ligado
        FimMetodo
        privado Metodo getIntensidade()
                retorne intensidade
        FimMetodo
        publico Metodo setLigado(l:logico)
                ligado = l
        FimMetodo
        publico Metodo setIntensidade(i:Inteiro)
                intensidade = i
        FimMetodo

FimClasse
```

Orientação à objetos

Pilares da POO: Encapsulamento

Exemplo



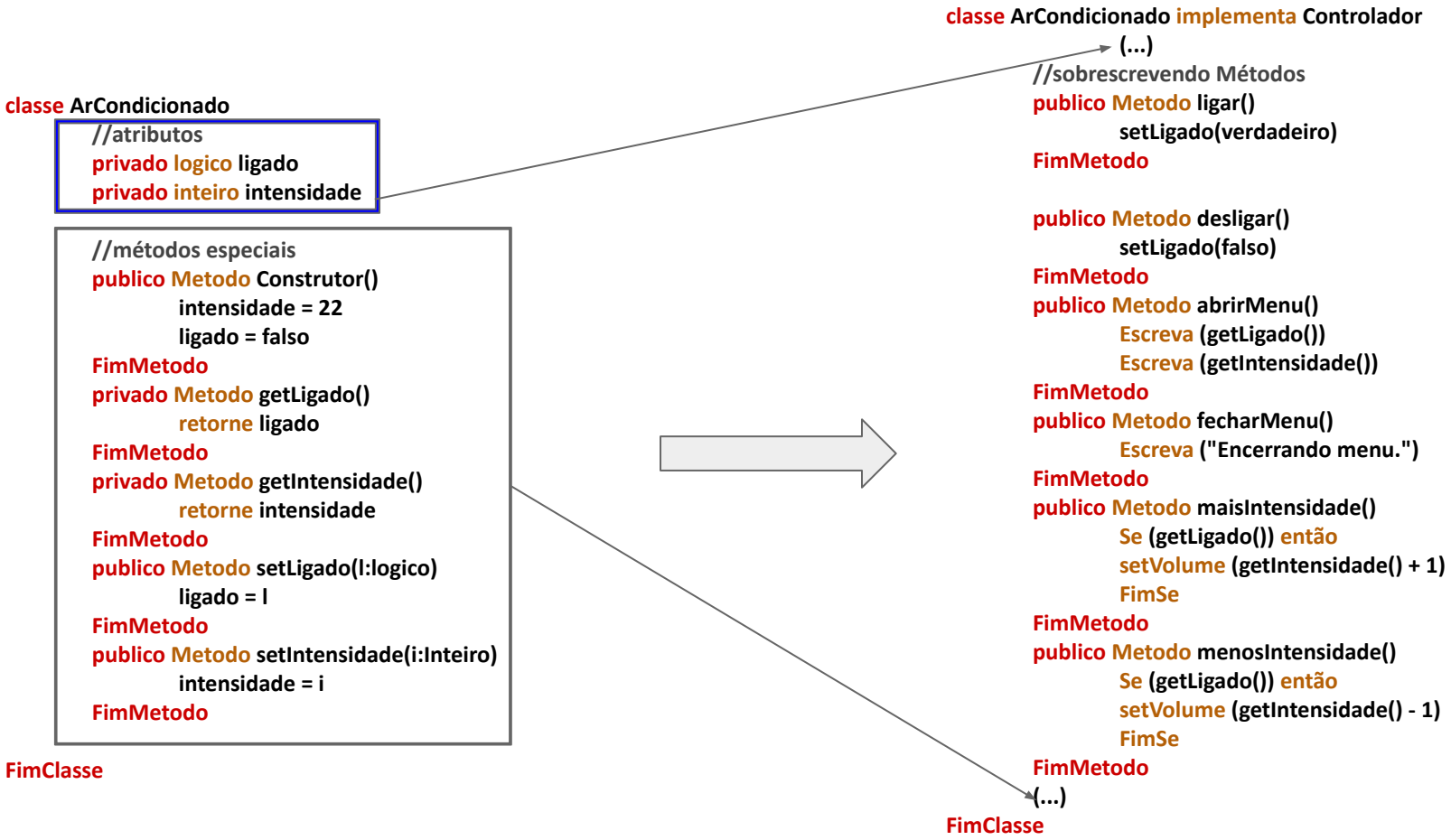
classe ArCondicionado implementa Controlador

FimClasse

- Classe "ArCondicionado" implementa a interface "Controlador".

Michelle Karissa Luciano Carvalho

UFBA Universidade Federal da Bahia computação UFBA


classe ArCondicionado
//atributos
privado logico ligado
privado inteiro intensidade
//métodos especiais
publico Metodo Construtor()
intensidade = 22
ligado = falso
FimMetodo
privado Metodo getLigado()
retorne ligado
FimMetodo
privado Metodo getIntensidade()
retorne intensidade
FimMetodo
publico Metodo setLigado(l:logico)
ligado = l
FimMetodo
publico Metodo setIntensidade(i:Inteiro)
intensidade = i
FimMetodo
FimClasse
classe ArCondicionado implementa Controlador
(...)
//sobrescrevendo Métodos
publico Metodo ligar()
setLigado(verdadeiro)
FimMetodo
publico Metodo desligar()
setLigado(falso)
FimMetodo
publico Metodo abrirMenu()
Escreva (getLigado())
Escreva (getIntensidade())
FimMetodo
publico Metodo fecharMenu()
Escreva ("Encerrando menu.")
FimMetodo
publico Metodo maisIntensidade()
Se (getLigado()) então
setVolume (getIntensidade() + 1)
FimSe
FimMetodo
publico Metodo menosIntensidade()
Se (getLigado()) então
setVolume (getIntensidade() - 1)
FimSe
FimMetodo
(...)
FimClasse

```
classe ArCondicionado
    //atributos
    privado logico ligado
    privado inteiro volume
    privado inteiro intensidade

    //métodos especiais
    publico Metodo Contrutor()
            volume = 50
            ligado = falso
    FimMetodo
    privado Metodo getLigado()
            retorne ligado
    FimMetodo
    privado Metodo getVolume()
            retorne volume
    FimMetodo
    privado Metodo getIntensidade()
            retorne intensidade
    FimMetodo
    publico Metodo setLigado(l:logico)
            ligado = l
    FimMetodo
    publico Metodo setVolume(v:Inteiro)
            volume = v
    FimMetodo
    publico Metodo setIntensidade(i:Inteiro)
            intensidade = i
    FimMetodo
FimClasse
```

```
classe ArCondicionado implementa Controlador
        (...)
    //sobrescrevendo Métodos
    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 ("|")
            FimPara
    FimMetodo
    publico Metodo fecharMenu()
            Escreva ("Encerrando menu.")
    FimMetodo
    publico Metodo maisVolume()
            Se (getLigado()) então
            setVolume (getVolume() + 1)
            FimSe
    FimMetodo
    publico Metodo menosVolume()
            Se (getLigado()) então
            setVolume (getVolume() - 1)
            FimSe
    FimMetodo
    (...)
FimClasse
```

Continuar…

Atividade 4

- **Finalizar o pseudo-código da classe "ArCondicionado" se houver necessidade.**
- **Inserir as classes abaixo no diagrama de classes e pseudo-código:**
    - **"Televisao"**
    - **"Iluminacao"**
    - **"Alarme"**

Objetos




Michelle Karissa Luciano Carvalho