



High performance. Delivered.

Application Delivery Fundamentals: Java

Module 7: Encapsulation

Module Objectives

- No final deste módulo, os participantes serão capazes de:
 - Descreva o princípio de encapsulamento OOP.
 - Explique o uso de modificadores de acesso no suporte ao encapsulamento.
 - Demonstrar considerações comuns de design de classe que suportam encapsulamento.



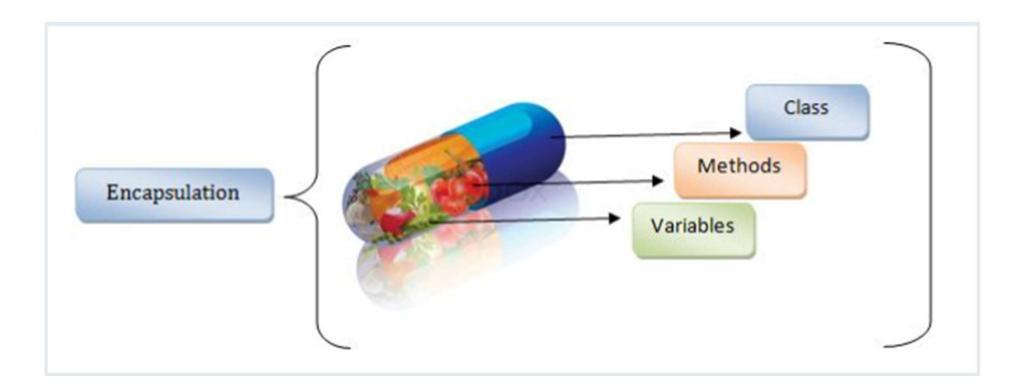
Encapsulamento

"Encapsulamento"

Encapsulamento

Encapsulation in Java





Encapsulation

- Encapsulamento é a ligação e ocultação de dados implementando-os como uma única entidade (uma classe).
- É usado para ocultar os valores ou o estado de um objeto dentro de uma classe, impedindo o acesso direto a eles pelos clientes de uma forma que possa violar a o estado.
- O encapsulamento reduz a complexidade de um sistema, separando as preocupações da interface de um objeto da sua implementação.

Access Control

- Uma classe encapsulada não deve expor detalhes de sua implementação a outros objetos.
- Um objeto externo n\u00e3o precisa saber os detalhes da implementa\u00e7\u00e3o de um objeto para enviar mensagens ao objeto.
- Essas interfaces públicas devem garantir que as modificações feitas no estado de um objeto sigam o design pretendido da classe desse objeto.

Access Modifiers

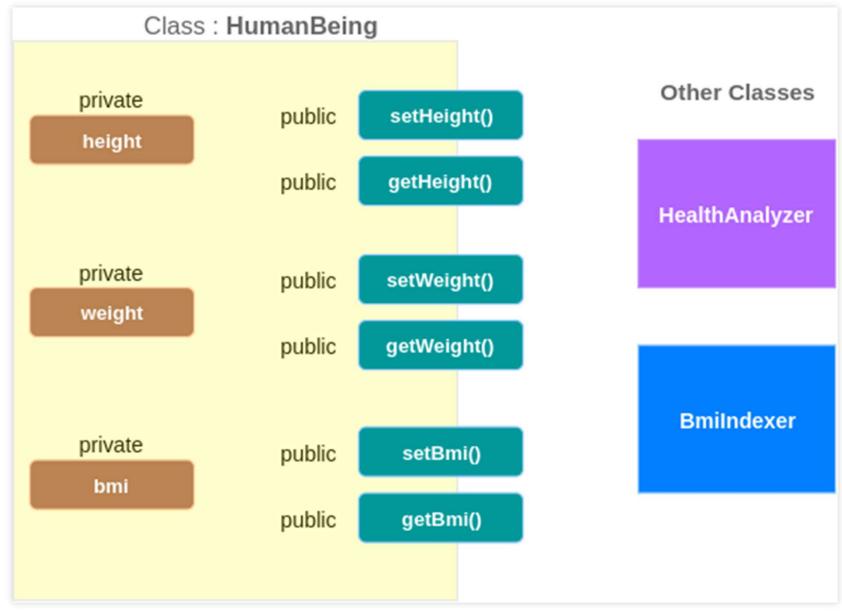
1. Class Access modifiers – describes how a class can be accessed

Modifier	Description
(no modifier)	A classe só pode ser acessada a partir do mesmo package
public	A classe pode ser acessada de qualquer lugar

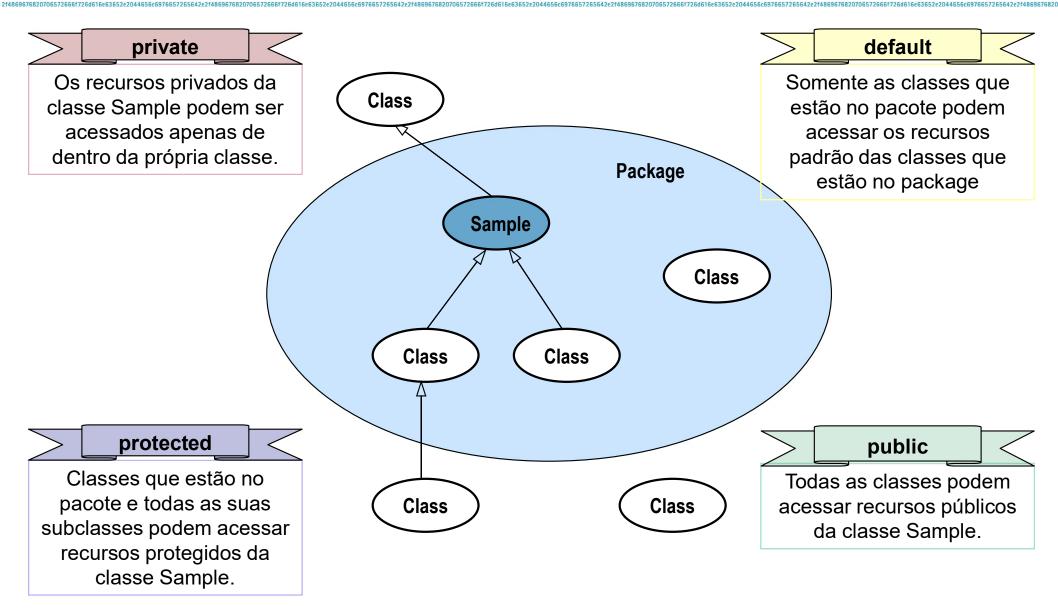
2. **Member Access** modifiers – describes how a member can be accessed

Modifier	Description
(no modifier)	Membro é acessível apenas dentro do seu package
public	O membro é acessível a partir de qualquer classe de qualquer package
protected	O membro é acessível em seu package de classe e por suas subclasses
private	O membro é acessível apenas a partir de sua classe

Encapsulamento



Member Access Modifiers Diagram



^{*} Default is not a modifier; it is just the name of the access level if no access modifier is specified.

'Setter' Methods

- Os campos ou o estado de um objeto geralmente são implementados por atributos privados;
- Para modificar atributos privados, os objetos apresentam interfaces públicas chamadas métodos "setter" ou "mutator";
- Nem todos os atributos privados exigem métodos de configuração; isso deve depender do design;
- Esses métodos devem controlar estritamente como os campos são modificados e executar validações apropriadas nos parâmetros passados;
- Sempre que os atributos de um objeto são modificados, o objeto deve validar a correção de seu estado (invariantes).

'Getter' Methods

- Os métodos "Getter" ou "accessor" permitem que objetos retornem os valores de seus atributos particulares.
- Nem todos os atributos privados precisam ter métodos getter; isso depende do design.
- Os métodos Getter sempre devem retornar apenas uma cópia dos valores dos atributos, e não os próprios atributos.

Refer to the SetterGetterSample.java, PersonSampleOne.java and PersonSampleTwo.java sample codes.

Perguntas:

O que é encapsulamento?

Significa separar o programa em partes, o mais isolado possível

O que é modificador de acesso?

São palavras-chave que garantem níveis de acesso aos atributos, métodos e classes

Para que servem os metodos Get e Set?

Eles servem para pegarmos informações de variáveis da classe que são definidas como 'private',





True or False?

- 1. O encapsulamento separa a preocupação da interface de um objeto com sua implementação. Isso, no entanto, aumenta a complexidade. <u>False</u>.
- 2. É sempre uma boa prática acessar ou alterar atributos privados e públicos da classe usando getters e setters. False.
- Encapsulamento refere-se à ligação de dados (usando classes) e ocultando-os do mundo externo (implementando interfaces).
 True
- 4. O modificador de acesso protected permite que os membros sejam acessíveis em seu pacote de classe e por suas subclasses. <u>True</u>

Questions and Comments

 What questions or comments do you have?



