Programação Orientada a Objetos em Java

Encapsulamento

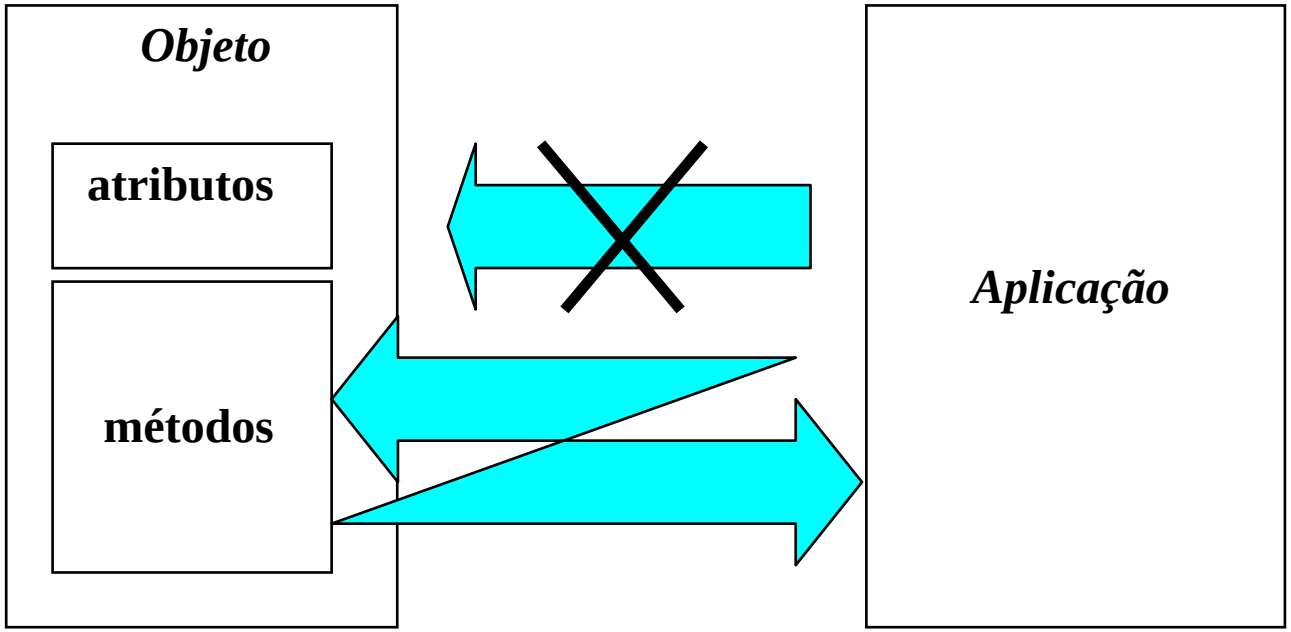
Prof. Henrique Louro
henrique.louro@cps.sp.gov.br

Cetec
Unidade de Ensino Médio e Técnico

CPS
Centro Paula Souza

GOVERNO DO ESTADO DE SÃO PAULO

Encapsulamento



Encapsulamento

- **Encapsulamento ou "*data hiding*" é um conceito bastante importante em orientação a objetos;**

- **É utilizado para restringir o acesso às declarações de uma classe e à própria classe.**

Encapsulamento
Modificadores de Visibilidade

- **public:** A classe, método ou atributo, são sempre acessíveis em todos os métodos de todas as classes;
- **private:** A classe, método ou atributo, são acessíveis somente por métodos da própria classe. É o nível mais rígido de encapsulamento;
- **protected:** A classe, método ou atributo, são acessíveis nos métodos da própria classe e suas subclasses (herança);
- **pakage:** A classe, método ou atributo, são acessíveis somente por métodos das classes que pertencem ao mesmo “pakage”.

Encapsulamento

- Em Java, a visibilidade padrão de classes, atributos e métodos está restrita a todos os membros que fazem parte de um mesmo pacote;
- A palavra-chave ***public*** modifica essa visibilidade de forma a ampliá-la, deixando-a sem restrições;
- Uma classe definida como pública pode ser utilizada por qualquer objeto de qualquer pacote;
- Em Java, uma unidade de compilação (um arquivo fonte com extensão ***.java***) pode ter no máximo uma classe pública, cujo nome deve ser o mesmo do arquivo (sem a extensão).
- As demais classes na unidade de compilação, não públicas, são consideradas classes de suporte para a classe pública e têm a visibilidade padrão.

Encapsulamento

- Um **atributo público** de uma classe pode ser diretamente acessado e manipulado por objetos de outras classes.
- Um **método público** de uma classe pode ser aplicado a um objeto dessa classe a partir de qualquer outro objeto de outra classe.
- O **conjunto de métodos públicos** de uma classe determina o que pode ser feito com objetos da classe, ou seja, determina o seu comportamento.
- A palavra-chave ***private*** restringe a visibilidade do membro modificado, método ou atributo, exclusivamente a objetos da própria classe que contém sua definição.

Encapsulamento

package (Pacote)

- No desenvolvimento de pequenas atividades ou aplicações, é viável manter o código e suas classes no diretório corrente;
- No entanto, para grandes aplicações é preciso organizar as classes de maneira a:
  - 1. evitar problemas com nomes duplicados de classes, e
  - 2. localizar o código da classe de forma eficiente.
- Em Java, a solução para esse problema está na organização de classes e interfaces em pacotes. Essencialmente, uma classe ***Tal*** que pertence a um pacote ***nome.do.pacote*** tem o nome completo ***nome.do.pacote.Tal*** e o compilador Java espera encontrar o arquivo ***Tal.class*** em um subdiretório ***nome/do/pacote***. Este, por sua vez, deve estar localizado sob um dos diretórios especificados na variável de ambiente **CLASSPATH**.

Encapsulamento

Getters e Setters

- Na linguagem Java, há uma convenção de nomenclatura para os métodos que têm como finalidade acessar ou alterar as propriedades de um objeto.

- Segundo essa convenção, os nomes dos métodos que permitem a consulta das propriedades de um objeto devem possuir o prefixo ***get***. Analogamente, os nomes dos métodos que permitem a alteração das propriedades de um objeto devem possuir o prefixo ***set***.

- Na maioria dos casos, é muito conveniente seguir essa convenção, pois os desenvolvedores Java já estão acostumados com essas regras de nomenclatura e o funcionamento de muitas bibliotecas do Java depende fortemente desse padrão.

Encapsulamento

Getters e Setters

- ***void setAtributo(tipo var)***: método utilizado para alterar o conteúdo de um atributo de uma classe, encapsulado com *private;*

- ***tipo getAtributo()***: método utilizado para acessar o conteúdo de um atributo de uma classe, encapsulado com *private*;

Encapsulamento

Getters e Setters: Exemplo classe Pessoa

```
public class Pessoa{
    private String nome;
    private int idade;
    Pessoa(String nome, int idade) {
        this.nome = nome;
        this.idade = idade;
    }
    public void imprimir() {
        System.out.println(this.nome + " " + this.idade);
    }
    public String getNome() {
        return nome;
    }
    public void setNome(String nome) {
        this.nome = nome;
    }
    public int getIdade() {
        return idade;
    }
    public void setIdade(int idade) {
        this.idade = idade;
    }
}
```

Encapsulamento

Getters e Setters: Exemplo classe executora

```
public class Testes {

    public static void main(String[] args) {
        Pessoa p = new Pessoa();

        p.setNome("Ana");
        p.setIdade(18);

        p.imprimir();

        System.out.println(p.getNome());
    }
}
```

Dúvidas?

Poste suas dúvidas preferencialmente no **Fórum de Dúvidas/Sugestões** da capacitação, na plataforma AVA.

Obrigado!

Prof. Henrique Louro
henrique.louro@cps.sp.gov.br