



High performance. Delivered.

Application Delivery Fundamentals: Java

Module 5: Object-Oriented Design and Principles

Module Objectives

- No final deste módulo, os participantes serão capazes de:
 - Definir o conceito de abstração de software em linguagens orientadas a objetos
 - Descrever os benefícios de uma abordagem externa ao projetar classes
 - Explicar os princípios de alta coesão e acoplamento flexível no design de objetos



Activity – Create a Java File

- Nessa atividade você deve:
 - Crie uma classe java Gato.java dentro do package sef.module5.sample
 - Com os atributos: nome e idade
 - Com os métodos: emitirSom;
 - No método constructor inicialize os atributos (não engessar os atributos);



Activity – Create a Java File

- Nessa atividade você deve:
 - Crie uma classe java Dog.java dentro do package sef.module5.sample
 - Com os atributos: nome e idade
 - Com os métodos: emitirSom;
 - No método constructor inicialize os atributos (não engessar os atributos);



Activity – Create a Java File

- Nessa atividade você deve:
 - Crie uma classe principal java
PrincipalAnimais.java dentro do package
sef.module5.sample com o método main
 - Instancie a classe Gato e a classe Dog;
 - Execute os métodos emitirSom;
 - O Gato faz “miau”;
 - O Dog faz “auau”;



Pilares da O.O.



Abstração

“Abstração”

Software Abstraction

- Abstração é o processo de redução ou simplificação de informações ou de um conceito para manter o conteúdo relevante para o contexto atual
- Os dados são abstraídos por construções de linguagem de alto nível, como:
 - Numbers
 - Letters
 - Higher level data concepts (Date, String)
- Funções e métodos são exemplos de abstração, uma vez que só vemos os nomes e os argumentos passados, não vemos o algoritmo que esta dentro do método;

Object-Oriented Programming (OOP)

- A orientação a objetos é um meio poderoso de expressar um domínio do problema e sua solução, mas ainda está sujeito a limitações e armadilhas.
- A mentalidade de um projetista de OO deve ser focada na criação de componentes reutilizáveis pequenos e simples que podem ser combinados para criar sistemas mais complexos.

Abstraction and Object-Oriented Programming (OOP)

Uma linguagem OOP é criada no torno de uma entidade de software chamada objeto

Um objeto é definido como a soma de dados, das funções e o comportamentos expostos por meio de uma interface pública.

Os objetos se comunicam através do uso de interfaces públicas e não precisam conhecer o funcionamento interno de outros objetos

Os objetos são baseados em classes, usadas para definir os atributos e comportamentos comuns a todas as instâncias dessa classe

Classes têm hierarquias que definem abstrações em suas classificações

High Cohesion

- É importante lembrar que um objeto une dados e comportamentos relevantes.
- Uma armadilha comum no OO é projetar um objeto para poder fazer muitas coisas, possivelmente não relacionadas ou pouco relacionadas (baixa coesão).
- Um objeto exhibe alta coesão se todo o seu comportamento estiver fortemente relacionado ao contexto da funcionalidade geral do objeto.
- A funcionalidade do próprio objeto deve ser projetada com um objetivo muito focado e específico.

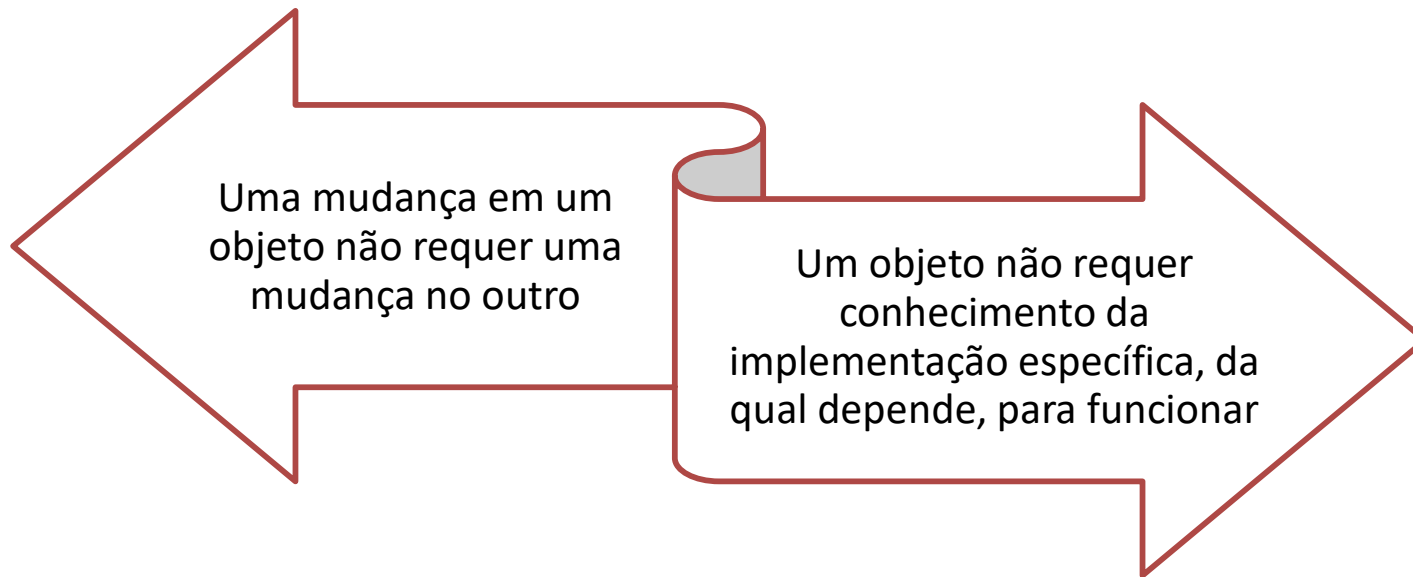
High Cohesion

Coesão
Qual é a sua função?



Loose Coupling

- Um objeto é fracamente acoplado à outro objeto quando :



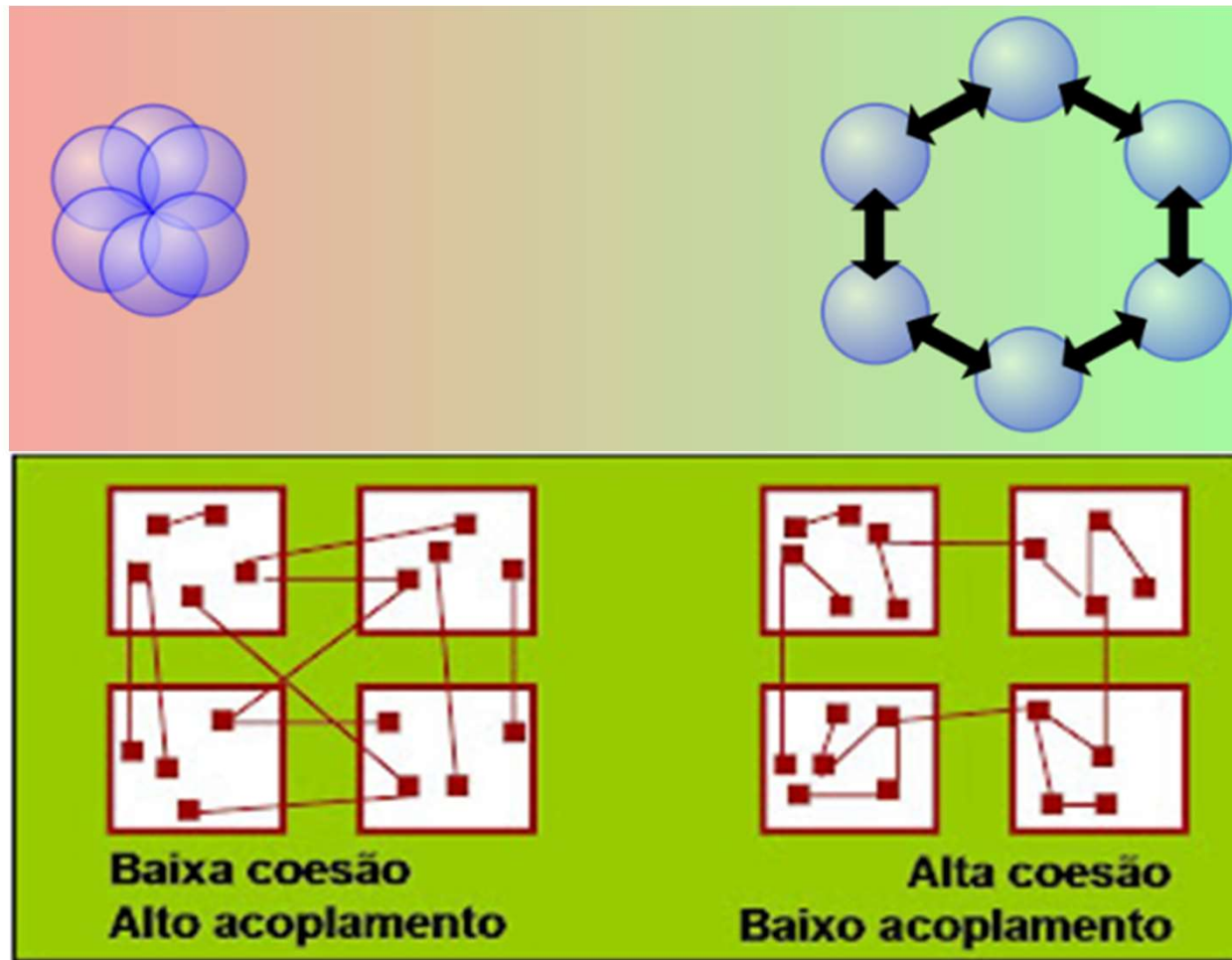
Engenharia de Software

- **Lembrar:** prezar pela **alta coesão** e **fraco acoplamento**.
- Uma das maneiras de se conseguir **fraco acoplamento** é com **interfaces**.
- Coesão e acoplamento são conceitos interligados: **classes coesas tendem a gerar baixo acoplamento**.



6

Engenharia de Software



Perguntas:

O que é uma abstração de software?

Abstração é o processo de redução ou simplificação de informações ou de um conceito para manter o conteúdo relevante para o contexto atual.

O que é um orientação a objetos?

"O termo orientação a objetos significa organizar o mundo real como uma coleção de objetos que incorporam estrutura de dados e um conjunto de operações que manipulam estes dados. "

O que é alta coesão?

Uma classe deve ter apenas uma única responsabilidade e realizá-la de maneira satisfatória, ou seja, uma classe não deve assumir responsabilidades que não são suas.

O que é um Baixo Acoplamento?

É o grau de dependências entre classes.

Padrões de Projeto:

- **Ligados a O.O.**
- **Soluções de problemas recorrentes já prontas testadas e aprovadas;**
- **Singleton**
 - Garante a existência de apenas uma instância de uma classe
- **Adapter**
 - Permite que classes com interfaces incompatíveis trabalhem juntas.
- **Factory**
 - Chamam o método fabrica especificado numa interface e implementado por um classe filha

Questions and Comments

- What questions or comments do you have?

