

02-OO-Fundamentos

Conteúdo: Paradigma Procedural x OO.
Princípios de OO. Objeto. Classe. Atributo.
Operação. Notação UML. Abstração.
Encapsulamento. Acoplamento. Coesão.
Herança. Polimorfismo. Classes Concretas x
Abstratas. Interface.

Bibliografia Recomendada

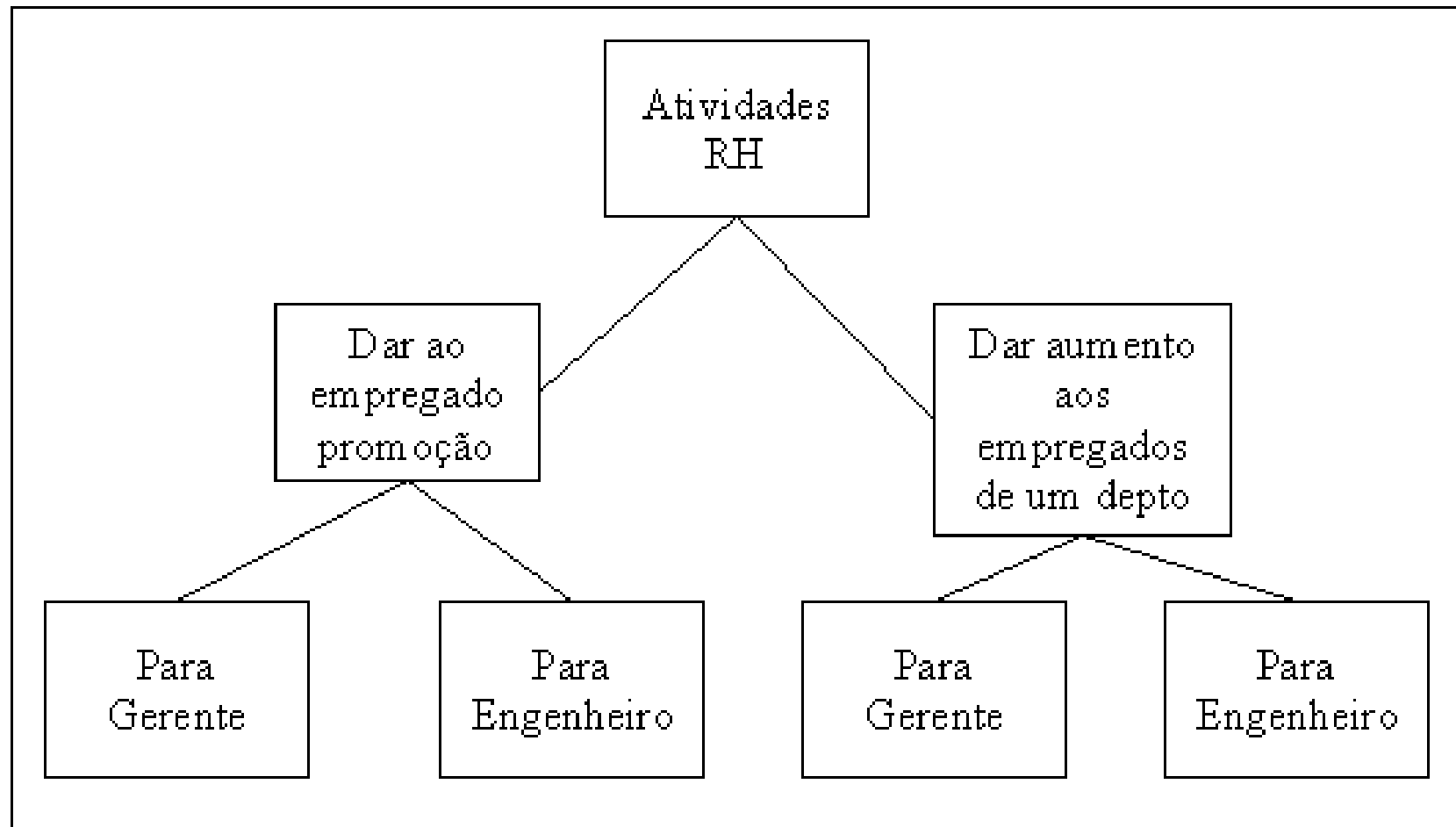
- Use a Cabeça! Java. 2ª edição. 2005. Kathy Sierra, Bert Bates. Editora Alta Books.



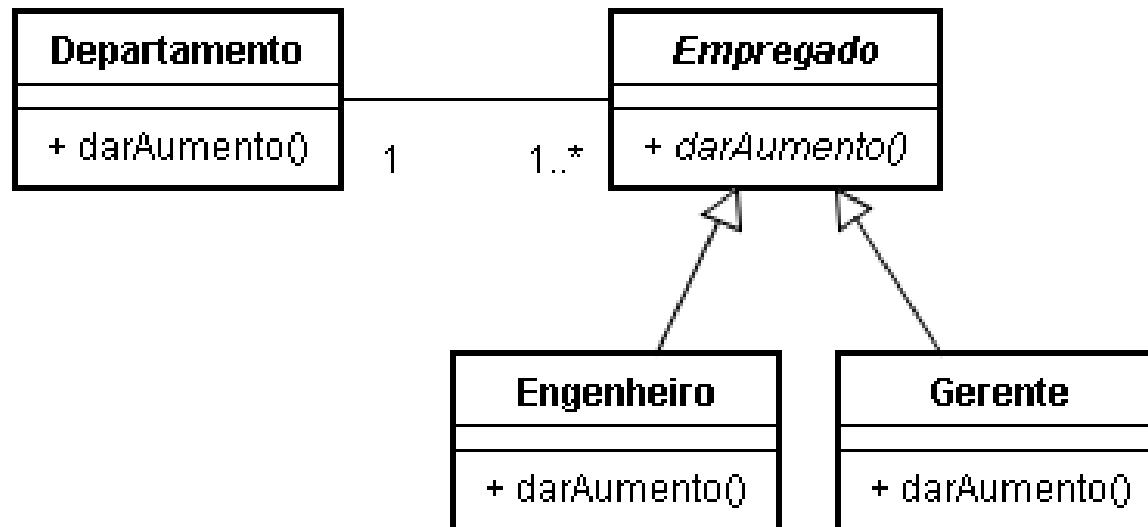
Paradigma Procedural x OO (1)

	Procedural	Orientação a Objetos
Foco	Verbo	Substantivo
Estrutura	Hierarquia	Rede de
Quantitacional	Tarefa/subtarefa	Colaboração
Manutenibilidade	Difícil	Robusta
Reuso	Copiar/Colar	Herança
	Uso de parâmetros	Composição de objetos
Config. em casos especiais	If / switch	Polimorfismo

Paradigma Procedural x OO (2)



Paradigma Procedural x OO (3)



Teste Rápido

- Considere a afirmação:
 - “O paradigma OO é superior ao paradigma procedural pois garante o reúso.” V ou F, justificando.

Resposta do Teste Rápido

- Falso. Embora a OO ofereça mais recursos para o reúso, como a herança e a composição de objetos, não há garantia que se fará reúso a menos que um bom projeto de software seja realizado.

Princípios OO (1)

- O que é Objeto?
 - Entre as várias definições, para o fim de POO podemos destacar: “Algo perceptível, conhecido, pensado ou representado, por qualquer um dos nossos sentidos, em caráter estável”
 - Pode ser algo físico (um carro, uma caneta, um sapo, etc.) ou algo puramente mental (uma conta bancária, uma equação, uma sessão de aula, etc.)

Princípios OO (2)

- Pensar
 - Substantivo → características
 - Verbos → comportamento
- Orientação a Objetos
 - Procura imitar na estrutura do programa a forma como enxergamos os elementos do contexto-problema observado
 - O mundo é percebido por nós como tendo "coisas", coisas essas que têm características e comportamentos → objetos, que interagem entre si

Princípios OO (3)

- Classe

- Abstração do objeto do mundo real, como um modelo simplificado para ser representado no computador
- Definem características e comportamentos afins a todos os objetos dessa classe
- A partir de uma classe, podemos criar 1 ou mais objetos (instanciação)

- Objeto

- Instância de um classe

Princípios OO (4)

- Propriedade
 - Característica comum a todos os objetos daquela classe.
- Operação
 - ação que objetos de uma classe podem executar. Um método público é uma implementação de uma operação.
- Mensagem: ocorre quando um objeto (emissor), durante a execução de um método, invoca o método de outro objeto (receptor).

Teste Rápido

- Cenário: “Um sistema de biblioteca realiza empréstimos de exemplares de livros a seus utilizadores. A partir do sétimo dia, o sistema envia diariamente um email alertando sobre o vencimento para devolução dos exemplares que estão de posse do utilizador, até que ele devolva.”
 - Identifique os elementos do modelo OO envolvidos nesse cenário.

Encapsulamento e Acoplamento (1)

- Objetos precisam colaborar entre si
- Interface do objeto: conjunto de mensagens que ele pode receber
- Encapsulamento
 - capacidade de esconder detalhes de implementação.

Encapsulamento e Acoplamento (2)

- Acoplamento
 - Quanto menos um objeto-usuário souber dos detalhes da implementação de um objeto receptor da mensagem, menos o objeto-usuário será afetado por eventuais alterações no receptor

Coesão

- Medida que determina o quão objetivo é o propósito de uma classe
- Quanto mais coesa, menor o nível de acoplamento com outras classes
- Princípio da Responsabilidade Singular
 - Single Responsibility Principle
 - Uma classe deve ter somente um motivo para mudar

Teste Rápido

- Um objeto armazena os itens comprados, calcula os impostos, gera a fatura em pdf, e realiza o envio de informações contábeis para o sistema de contabilidade. Que princípio de OO está sendo violado?

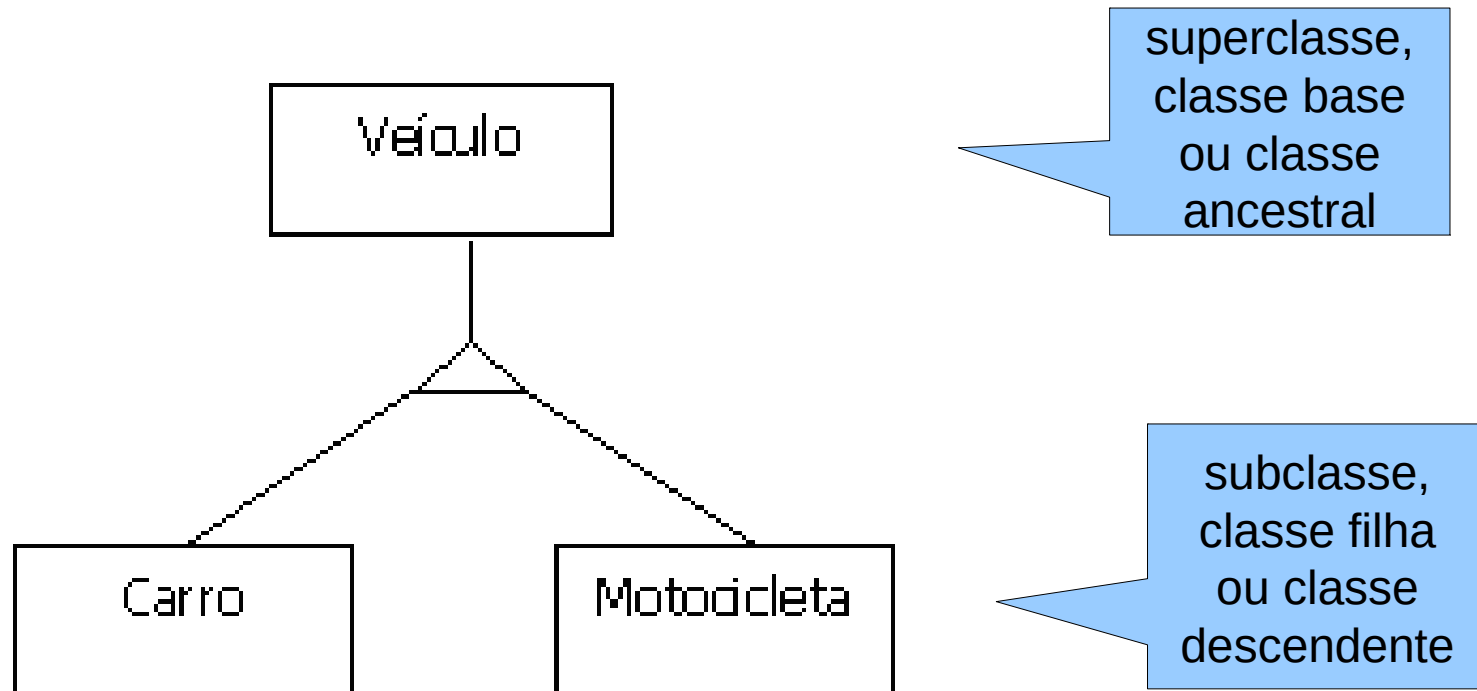
Resposta do Teste Rápido

- Coesão. Esse objeto está com muitas responsabilidades, apontando a necessidade de distribuição dessas responsabilidades para outros objetos.

Herança (1)

- Mecanismo de extensão oferecido pela OO
 - criação de classes baseadas em classes já existentes para conseguir uma reutilização de código segura.
 - É a capacidade de criarmos novos tipos de objetos (classes) a partir de tipos já existentes.
 - A ideia é especializar as características herdadas de uma classe mais genérica. A classe filha herda todas as características e comportamentos da classe pai. Isso cria o que se chama de árvore de hierarquia.

Herança (2)



Teste Rápido

- A herança quebra o encapsulamento. V ou F, e justifique.

Resposta do Teste Rápido

- Verdadeiro. O reuso por herança expõe, para a subclasse (ou filha), detalhes da implementação herdados da super classe (ou pai), portanto quebrando o encapsulamento. Entretanto, isso não invalida o uso da herança, contanto que seja aplicada com muito cuidado.
 - “Bons engenheiros de software seguem estritamente as regras, os excelentes sabem como e quando desrespeitá-las.”

Polimorfismo (1)

- Origem grega: significa "muitas formas" (poli = muitas, morphos = formas)
- Nasce da separação da interface de objeto das implementações dessa interface
- Uma mesma mensagem que pode ser enviada para vários objetos, comporta-se de maneira diferente dependendo da implementação definida na classe daquele objeto.



Escreva um resumo (1 pág. no máximo) sobre polimorfismo.
Consultar wikipédia

Tarefa Individual

- Ler o capítulo 1 do livro do GoF
 - Padrões de Projeto – Soluções reutilizáveis de software orientado a objetos.
 - Capítulo disponível para leitura gratuita na internet, pelo Google Books.
 - <https://play.google.com/books/reader?id=U91CYCqTCgkC&hl=en&pg=GBS.PA17>



Humor

