

PROGRAMAÇÃO ORIENTADA A OBJETOS

Prof. Edkallenn

Sistemas de Informação - Uninorte



Programação Orientada a Objetos

- **Prof. Edkallenn Lima**
- edkallenn@yahoo.com.br (somente para dúvidas)
- **Blogs:**
 - <http://professored.wordpress.com> (Computador de Papel – O conteúdo da forma)
 - <http://professored.tumblr.com/> (Pensamentos Incompletos)
 - <http://umcientistaporquinzena.tumblr.com/> (Um cientista por quinzena)
 - <http://eulinoslivros.tumblr.com/> (Eu li nos livros)
 - <http://linabiblia.tumblr.com/> (Eu li na Bíblia)
- **YouTube:**
 - <https://www.youtube.com/user/edkallenn>
 - <https://www.youtube.com/channel/UC-pD2gnahhxUDVuTAA0DHoQ>
- **Redes Sociais:**
 - <http://www.facebook.com/edkallenn>
 - <http://twitter.com/edkallenn> ou @edkallenn
 - <https://plus.google.com/u/0/113248995006035389558/posts>
 - Instagram: <http://instagram.com/edkallenn> ou @edkallenn
 - Foursquare: <https://pt.foursquare.com/edkallenn>
 - Pinterest: <https://br.pinterest.com/edkallenn/>
- **Telefones:**
 - 68 98401-2103 (CLARO e Whatsapp) e 68 3212-1211.

Os exercícios devem ser enviados SEMPRE para o e-mail:
edkevan@gmail.com
ou para o e-mail:
edkallenn.lima@uninorteac.com.br



Agenda

- Identidade de objetos
- Escopo
- Modificadores de acesso
- Métodos getters e setters



A identidade (igualdade) de um objeto

- Por definição, todo objeto é **único**.
- Se tivermos uma classe e forem criados dois objetos a partir dela, cada um será diferente do outro, **mesmo que seus estados sejam iguais por coincidência**.
- Porém, cada sistema terá necessidades específicas para definir o que torna um objeto igual a outro.
- Devido a isso, **a identidade ou igualdade de objetos deve ser definida por quem criou a classe**, pois só este tem o conhecimento do contexto em questão para poder determinar o que torna dois objetos iguais



A identidade (igualdade) de um objeto

- Poder determinar se dois objetos são iguais é de grande utilidade em sistemas, pois se uma das premissas deles é automatizar um processo do mundo real, nada melhor do que **poder prover facilidades que agilizem a execução das atividades.**
- Por exemplo, imagine um estoque de uma loja de eletrodomésticos. Se um funcionário for precisar encontrar um determinado produto, em um processo manual ele iria se dirigir ao estoque e visualmente procurar estante por estante, prateleira por prateleira
- Para poder efetivamente verificar se um objeto é igual ao outro, devemos utilizar um método, pois, como já vimos, são estes os responsáveis por definirem os comportamentos, ações das classes/objetos e, neste caso, o comportamento que torna dois objetos iguais



A identidade (igualdade) de um objeto

- Em Java e C# , existe um método específico para determinar se dois objetos são iguais: o **equals** .
- Como a função deste é **determinar se dois objetos são iguais**, nada mais óbvio que ele retorne um **booleano**.
- É a partir da definição deste método que podemos definir o que torna dois objetos iguais



Representação numérica de um objeto

- Embora não seja um conceito de fato da Orientação a Objetos, a representação numérica de um objeto (no caso, seu *hashcode*) é muito útil, e linguagens como Java , C# , Ruby , entre outras disponibilizam tal representação.
- Basicamente, um *hashcode* é um número que é gerado a partir do estado corrente do objeto, ou seja, dos valores de seus atributos naquele determinado momento.
- Este número serve para otimizar e pesquisar em estruturas que utiliza tables hash (tabelas de dispersão).
- Não é o objetivo deste curso explicar tables hash, mas saber de sua existência é o primeiro passo para seu uso de forma eficiente.



Escopo e modificadores de acesso

- O **escopo** dos campos e variáveis dentro de uma classe determina a sua **visibilidade** (isto é, se as variáveis ou campos podem ser acessadas ou modificadas em todos os métodos da classe, somente um determinado método ou mesmo somente em parte de um determinado método).

Qualificador	Descrição
public	Define que o conteúdo da classe é público e pode ser utilizado livremente por outras classes do mesmo pacote ou de outro pacote.
protected	Define que o conteúdo da classe está protegido e que só pode ser utilizado por classes do mesmo pacote.
private	Define que o conteúdo é privado e só pode ser utilizado internamente na própria classe.



Identificador this

- O identificador `this` é responsável por fazer **referência a um atributo ou método da própria classe**.
- Isso é muito útil, pois **é comum que os parâmetros possuam nomes iguais aos já existentes** nos atributos de uma classe, diferenciando-os
- Uma alternativa para a utilização do identificador *this* é sua definição nas classes Java para a chamada de um método construtor dentro da própria classe.
- Isso é feito por meio do método reservado *this()*



Métodos get e set

- Existem métodos convencionados entre os programadores **Java que criam uma interface** entre os atributos e o usuário, sendo uma camada de comunicação pública.
- São denominados métodos **get** e **set** vinculados a cada atributo das classes e evitam o acesso direto a tais atributos, aumentando a segurança ao impedir que a implementação e a forma como os atributos são tratados estejam visíveis
- Nesta nova contextualização, **os atributos devem obrigatoriamente ser acessados por meio de métodos.**
- Os métodos **set** são utilizados para atribuir valores e os métodos **get** para a recuperação dos valores atuais dos atributos.
- Isso consiste em uma boa prática de programação que deve ser inserida em seu cotidiano, ainda mais no que tange à programação Java



Perguntas e Discussão

