

# Classes Internas (Java)

From Wiki\*\*3

Programação com Objectos

[Expand]

## Contents

- 1 Classe internas (não "static") em Java
- 2 Organização de código
- 3 Aspectos Básicos
- 4 Exemplos
  - 4.1 Gatos Iteráveis
  - 4.2 Comparadores em Java
  - 4.3 Gato e Estômago
  - 4.4 Carros e Motores
- 5 Exercícios

Na descrição mais simples, uma classe interna não é mais do que uma classe definida dentro de outra, sendo o acesso a ela controlado pela classe externa através das habituais especificações **public**, **private**, etc. Ainda nesta abordagem simples, este processo permite definir âmbitos restritos de visibilidade e acesso.

Em C++ e para classes internas Java declaradas **static**, a descrição "simples" é a que realmente é implementada pela linguagem. Nestes casos, serve apenas como processo de organização de código, não havendo necessariamente uma relação entre as classes internas e as externas que as contêm. Embora estes usos tenham utilização, especialmente em C++ (onde são a única possibilidade), não são a forma habitual de definir classes internas em Java.

## Classe internas (não "static") em Java

As classes internas em Java são mais que classes definidas dentro de outras classes. A definição destas classes implica que existe uma relação estreita entre uma classe interna e a externa que a contém. A definição é tal que as instâncias de classes internas têm acesso implícito às instâncias das classes externas, no conteúdo das quais foram criadas. Como se verá abaixo esta ligação é muito importante para os aspectos interessantes da programação utilizando este tipo de abstrações.

De modo análogo ao do acesso nos métodos de uma classe, o código de uma classe interna é considerado código da classe externa e pode ser por ela utilizada. Reciprocamente, o código da classe interna tem acesso implícito (directo) ao código e atributos da classe externa, mesmo que esta os tenha definido como **private**.

A criação de objectos com base em classes internas, na segunda definição, não é sequer possível sem a existência prévia de uma instância da classe externa. Os vários exemplos abaixo ilustram esta questão e introduzem novas formas de escrita (sintaxe) relacionadas com a criação e inicialização (**new** e construtores) destes objectos.

## Organização de código

- Organização de interfaces e classes e acesso ao código
- Classes internas locais (métodos e blocos)
- Classes anónimas

## Aspectos Básicos

- Classes internas e os vários aspectos de definição: classes internas simples; implementação de interfaces
- Ligação implícita entre objectos de classes internas e a instância da classe externa
- Partilha de acesso entre classe externa e interna
- Herança e classes internas e externas

## Exemplos

Estes exemplos ilustram o uso de classes internas, evidenciando vantagens relativamente à ocultação de pormenores de implementação e consequente melhoria na abstracção e organização de código.

## Gatos Iteráveis

O exemplo dos gatos iteráveis mostra como iterar um gato constituído por várias partes. Começa-se com uma implementação menos elegante e termina-se com a "melhor" implementação.

- Gatos Iteráveis -- como iterar um gato?

## Comparadores em Java

Os exemplos seguintes dizem respeito a interfaces Java.

- Comparações em Java -- definição de comparadores usando interfaces Java pré-definidas.

## Gato e Estômago

Os exemplos apresentados nos vários com um gato e um estômago são semelhantes aos anteriores, mostrando como localizar a definição de uma classe interna junto do código onde é utilizada.

- Gato e Estômago -- gatos e estômagos (várias configurações) (exemplo antigo -- ver iteradores acima)

## Carros e Motores

Os exemplos de carros e motores apresentam casos em que existe herança, tanto nas classes externas, como nas classes internas e casos mistos. Note-se que alguns destes casos não são "normais" e apenas se apresentam para evidenciar consequências relativas à programação com classes internas e herança.

- Carros e Motores -- aspectos e problemas de herança com carros e motores

## Exercícios

- Exercício 01: Interfaces `java.lang.Comparable<T>` e `java.util.Comparator<T>`
- Exercício 02: Iteração

Categories: **Ensino PO Java**