



SERVIÇO NACIONAL DE APRENDIZAGEM INDUSTRIAL

SENAI “GASPAR RICARDO JUNIOR”

Curso

**TÉCNICO EM DESENVOLVIMENTO
DE SISTEMAS**

Métodos equals e hashCode em Java

Matheus Almeida

Sorocaba
11 – 2024



SERVIÇO NACIONAL DE APRENDIZAGEM INDUSTRIAL

SENAI “GASPAR RICARDO JUNIOR”

Matheus Almeida

Métodos equals e hashCode em Java

Pesquisa sobre métodos equals e
hashCode em Java

Prof. – Emerson Magalhães

Sorocaba
11 – 2024

SUMÁRIO

INTRODUÇÃO.....	2
1. FUNDAMENTOS TEÓRICOS	2
2. UTILIZAÇÃO PRÁTICA EM COLEÇÕES JAVA E NO SPRING.....	3
3. LOMBOK: SIMPLIFICAÇÃO DO CÓDIGO.....	3

Métodos equals e hashCode em Java

INTRODUÇÃO

- Contextualização dos métodos equals e hashCode:

Os métodos equals e hashCode em Java desempenham um papel crucial na comparação e na organização de objetos em coleções baseadas em hashing, como HashMap e HashSet. Esses métodos ajudam a garantir que os objetos sejam comparados corretamente em termos de igualdade e que as coleções baseadas em hashing se comportem como esperado.

- Importância de equals e hashCode para coleções e frameworks como Spring:

Em coleções como HashMap e HashSet, os métodos equals e hashCode são usados para determinar se dois objetos são iguais e como eles devem ser armazenados ou recuperados. No Spring, esses métodos também têm relevância em operações de persistência e caching, especialmente quando se lida com entidades que precisam ser comparadas.

1. FUNDAMENTOS TEÓRICOS

- Explicação do contrato entre equals e hashCode:

O contrato entre equals e hashCode é fundamental para garantir a consistência de coleções baseadas em hashing. O contrato estabelece que se dois objetos são iguais, eles devem ter o mesmo código hash. Essa relação é importante para o correto funcionamento de coleções como HashMap.

- Regras que governam a implementação de equals e hashCode:

Para implementar corretamente esses métodos, devem ser seguidas algumas regras fundamentais, como reflexividade, simetria, transitividade, consistência e a garantia de que objetos iguais retornem o mesmo valor de hashCode.

- Como o contrato entre equals e hashCode afeta o comportamento das coleções (ex:

HashMap, HashSet):

Um comportamento incorreto na implementação pode resultar em objetos não sendo encontrados corretamente em coleções ou inseridos de maneira inesperada.

- Importância da implementação correta de equals e hashCode em entidades de aplicações Java:

A correta implementação desses métodos é essencial para que entidades sejam comparadas corretamente durante operações de persistência, caches, e outros processos críticos no Spring

2. UTILIZAÇÃO PRÁTICA EM COLEÇÕES JAVA E NO SPRING

Exemplos de código podem ser usados para ilustrar como esses métodos afetam o funcionamento das coleções.

```
1 public class Pessoa {
2     String nome;
3     String sobrenome;
4     String idade;
5     @Override
6     public boolean equals(Object obj) {
7         if (!(obj instanceof Pessoa))
8             return false;
9         Pessoa other = (Pessoa) obj;
10        if (!this.nome.equals(other.nome) || !this.idade.equals(other.idade))
11            return false;
12        return true;
13    }
14 }
```

Pessoa.java hosted with ❤ by GitHub

[view raw](#)

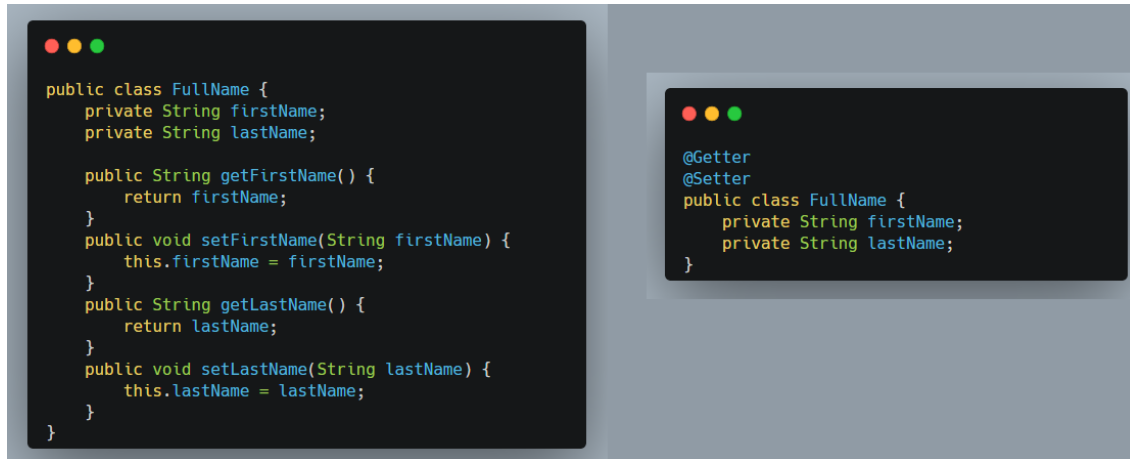
Com esse método garantimos que um objeto com mesmo nome e idade vai ser sempre considerado igual

3. LOMBOK: SIMPLIFICAÇÃO DO CÓDIGO

- Introdução à biblioteca Lombok:

O Lombok simplifica muito a implementação de código Java, oferecendo diversas anotações que quando adicionadas ao código, ensinam o compilador durante o processo de compilação a criar código Java.

- Exemplo prático de implementação de uma entidade com Lombok, comparando com uma implementação manual:



Adicionando as anotações **@Getter** e **@Setter** no topo da classe, dissemos ao Lombok que crie os métodos para todos os atributos da classe. Para criar um construtor vazio usamos a anotação **@NoArgsConstructor**.

- Vantagens e Desvantagens de Usar Lombok para equals e hashCode:

- Vantagens:

- Redução de código boilerplate.
- Melhor legibilidade e manutenção do código.
- Foco na lógica da aplicação.

- Desvantagens:

- Dependência de uma biblioteca externa.
- Potenciais problemas com depuração de código.
- Discussão sobre boas práticas de uso de Lombok em ambientes de produção:

CONCLUSÃO

Dessa forma, os métodos equals e hashCode são essenciais para garantir o funcionamento correto de coleções em Java e operações de persistência em frameworks como Spring. A biblioteca Lombok

facilita essa implementação, gerando automaticamente esses métodos e reduzindo o código repetitivo. No entanto, seu uso deve ser ponderado, considerando dependências externas e possíveis desafios de depuração.

No geral, o uso correto de equals, hashCode e Lombok permite desenvolver aplicações Java mais eficientes e com melhor manutenção, sendo uma prática recomendada para projetos que demandam alta escalabilidade e consistência no uso de coleções e entidade.

