SERVIÇO NACIONAL DE APRENDIZAGEM INDUSTRIAL

SENAI “GASPAR RICARDO JUNIOR”

Curso

TÉCNICO EM DESENVOLVIMENTO

DE SISTEMAS

Métodos equals, hashCode em Java e uso de Lombok

João Vittor Amorim da Silva

Sorocaba
Novembro – 2024

SERVIÇO NACIONAL DE APRENDIZAGEM INDUSTRIAL

SENAI “GASPAR RICARDO JUNIOR”

João Vittor Amorim da Silva

Métodos equals, hashCode em Java e uso de Lombok

Importância de equals e hashCode
para coleções e frameworks como Spring
Prof. Emerson Magalhães

Sorocaba
Novembro – 2024

SUMÁRIO

INTRODUÇÃO

Em Java, equals e hashCode são métodos herdados da classe Object, essenciais para determinar a igualdade e a localização de objetos em coleções. O método equals verifica se dois objetos são considerados iguais, enquanto hashCode produz um valor inteiro único para cada objeto, facilitando a busca em coleções baseadas em hashing, como HashSet e HashMap.

1. INTRODUÇÃO

1.1. Importância de equals e hashCode para coleções e frameworks como Spring

Coleções como HashSet e HashMap utilizam hashCode para alocar e acessar objetos de forma eficiente. Em frameworks como Spring, equals e hashCode são frequentemente usados para comparação e gestão de entidades persistentes e em cachingRegras que governam a implementação de equals e hashCode

1.2. Introdução ao Lombok

Lombok é uma biblioteca Java que reduz o "boilerplate" (código repetitivo) com anotações, gerando automaticamente métodos como equals, hashCode e toString. A anotação @EqualsAndHashCode é especialmente útil para classes que precisam implementar esses métodos, mantendo o código mais enxuto.

2. FUNDAMENTOS TEÓRICOS

2.1. Contrato entre equals e hashCode

O contrato de equals e hashCode estipula:

Se equals é sobrescrito, hashCode também deve ser.

Objetos iguais (obj1.equals(obj2) == true) devem ter o mesmo hash code.

Objetos com hash codes iguais não precisam ser iguais, mas objetos diferentes idealmente devem ter hash codes diferentes para evitar colisões.

2.2. Comportamento nas coleções (ex.: HashMap, HashSet)

Em coleções baseadas em hashing, como HashMap e HashSet, hashCode determina o “bucket” onde o objeto será armazenado. Se dois objetos têm o mesmo hash code, eles são colocados no mesmo bucket, mas equals é usado para verificar se são iguais. Uma implementação incorreta de equals e hashCode pode causar problemas de desempenho ou impedir que objetos sejam localizados corretamente.

2.3. Relevância em aplicações Java

Esses métodos são fundamentais para a consistência dos dados. Em frameworks de persistência como Hibernate (usado pelo Spring), a correta implementação desses métodos ajuda a garantir que o framework consiga identificar e manipular corretamente as entidades no banco de dados.

3. UTILIZAÇÃO PRÁTICA EM COLEÇÕES JAVA E NO SPRING

3.1. Exemplo com HashSet

```
Set<MyClass> mySet = new HashSet<>();
MyClass obj1 = new MyClass("value");
MyClass obj2 = new MyClass("value");
mySet.add(obj1);
System.out.println(mySet.contains(obj2));
```

3.2. Exemplo com uma Entidade Spring

```
@Entity
public class User {
    private Long id;
    private String name;

    @Override
    public boolean equals(Object o) { ... }

    @Override
    public int hashCode() { ... }
}
```

4. LOMBOOK: SIMPLIFICAÇÃO DO CÓDIGO

4.1. Introdução ao Lombok

Lombok permite a criação automática dos métodos equals e hashCode com a anotação @EqualsAndHashCode. Isso reduz o trabalho manual e torna o código mais limpo.

4.2. Comparação com Implementação Manual

4.2.1. Vantagens

- Menos código repetitivo;
- Fácil manutenção, pois elimina a necessidade de implementar manualmente os métodos.

4.2.2. Desvantagens

- Dependência externa: O uso de Lombok pode criar dependência em uma biblioteca externa.
- Depuração (debugging): O código gerado pelo Lombok é transparente, mas pode ser mais difícil de debugar diretamente, já que os métodos não estão visíveis no código fonte.

4.3. Boas Práticas

Lombok é recomendado em projetos de produção onde há suporte para gerenciar dependências, mas deve ser utilizado com cuidado em projetos onde a transparência e o controle dos métodos equals e hashCode são essenciais.

CONCLUSÃO

Os métodos equals e hashCode desempenham papéis cruciais na consistência e eficiência das coleções Java e dos frameworks de persistência como Spring. A biblioteca Lombok traz um grande benefício ao automatizar sua implementação, reduzindo o código boilerplate, mas traz também desafios como a dependência em uma biblioteca externa.

BIBLIOGRAFIA

Documentação oficial do Java para Object.equals e Object.hashCode.

Documentação da biblioteca Lombok para @EqualsAndHashCode.

Baeldung

DigitalOcean

CodeJava