



07. Equals e hashCode

📅 Date	@27/10/2022
📁 Categoria	Java
📖 Curso	Java Collections: Dominando Listas Sets e Mapas

Tópicos

- Equals e hashCode
- Considerando hashCode e Equals
- Verificando se está matriculado
- Programando defensivamente
- Para saber mais: O contrato do método equals

Considerando hashCode e Equals

- A classe String já possui um método próprio para gerar um hashCode e a não ser que estejamos trabalhando em um caso específico, podemos sempre usá-lo quando preciso.
- É uma boa prática - pra não dizer que é quase obrigatório - sempre que reescrevemos o método equals também reescrevermos o método hashCode, já que mesmo que no nosso código não utilize nenhum Set, existem diversos códigos que o utilizam, e caso não sobrescrevemos este método podemos esbarrar em bugs não esperados.
- Apesar de ser perigoso, se estamos verificando se um elemento pertence a uma implementação de **List**, só precisamos reescrever o método `equals()`, já que o método `.contains()` de List só utiliza o equals para comparação.

Para saber mais: O contrato do método equals

Nossa implementação do método `equals` é funcional, porém em alguns casos mais específicos podemos ter alguns problemas. Existe um contrato mais avançado que devemos seguir para implementar um método `equals` eficiente:

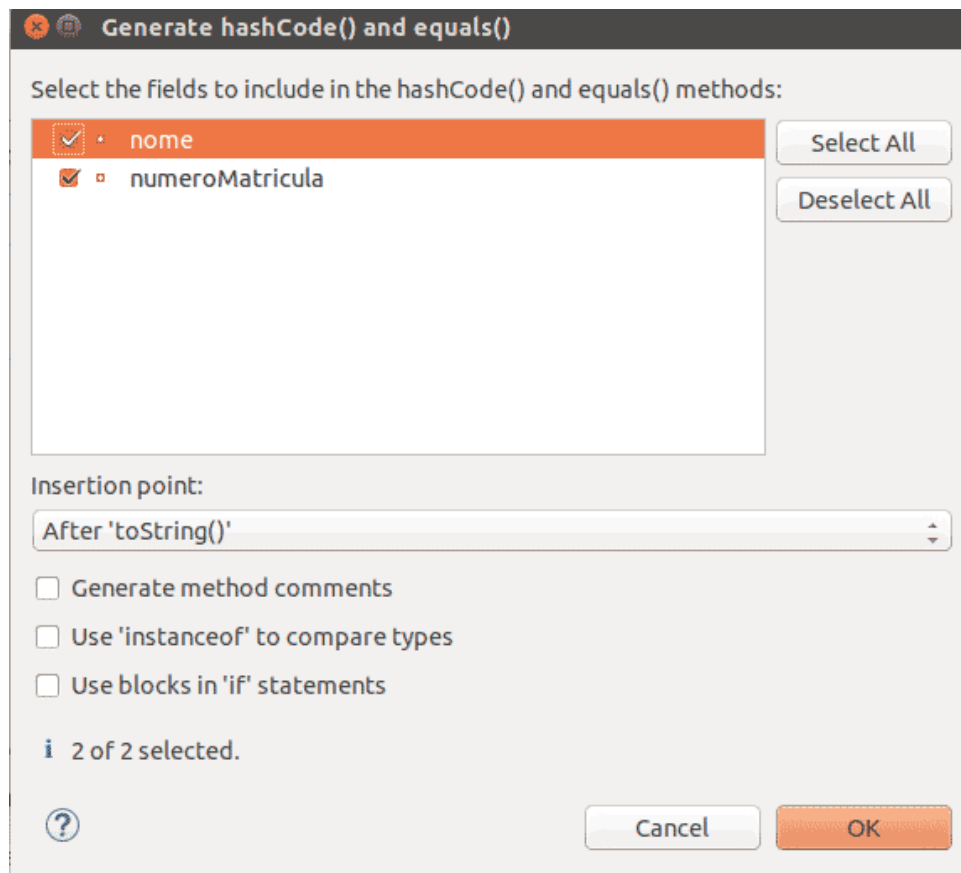
<https://docs.oracle.com/javase/8/docs/api/java/lang/Object.html#equals-java.lang.Object->

Por conta dessas propriedades uma implementação sofisticada do método `equals` pode ser bem trabalhosa, por essa razão que as IDE's fornecem recursos que implementam esse método para nós.



No Eclipse você pode pressionar **CTRL + 3** e digitar **equals**.

Podemos ainda escolher os atributos que queremos utilizar na comparação:



Sendo assim, utilize esse recurso para implementar os métodos *equals* e *hashCode* da classe `Aluno`.

A implementação deve ficar parecida com essa:

```
@Override
public int hashCode() {
    final int prime = 31;
    int result = 1;
    result = prime * result + ((nome == null) ? 0 : nome.hashCode());
    result = prime * result + numeroMatricula;
    return result;
}

@Override
public boolean equals(Object obj) {
    if (this == obj)
        return true;
    if (obj == null)
        return false;
    if (getClass() != obj.getClass())
```

```
        return false;
    Aluno other = (Aluno) obj;
    if (nome == null) {
        if (other.nome != null)
            return false;
    } else if (!nome.equals(other.nome))
        return false;
    if (numeroMatricula != other.numeroMatricula)
        return false;
    return true;
}
```
