**Comparação de Strings**

Em Java, é possível comparar duas Strings utilizando o operador ==. Porém, esse operador verifica apenas se as duas variáveis apontam para o mesmo objeto na memória, e não se o conteúdo das Strings é igual. Para comparar o conteúdo de duas Strings, é necessário utilizar o método equals(). Por exemplo:

String senha = "12345";

if (senha.equals("12345")) {

System.out.println("Acesso autorizado!");

} else {

System.out.println("Senha incorreta.");

}

Nesse caso, o método equals() é utilizado para comparar o conteúdo da variável senha com a String "12345". Se as duas Strings forem iguais, a mensagem "Acesso autorizado!" será impressa, caso contrário, a mensagem "Senha incorreta." será impressa.

Mais adiante trabalharemos também com o método equalsIgnoreCase(), que é usado para que a comparação de Strings desconsidere as letras maiúsculas e minúsculas. Utilizando apenas o equals, as String “alura” e “Alura” seriam consideradas diferentes.

## Text Block

Introduzido na versão 15 do Java, o Text Block é uma nova forma de representar Strings que facilitam a escrita de textos com múltiplas linhas. Em vez de utilizar aspas duplas para delimitar o texto e inserir quebras de linha manualmente, ou utilizar concatenações, é possível utilizar uma sintaxe mais simples que permite inserir o texto exatamente como ele é.

## Sintaxe do Text Block

Para criar um Text Block em Java, basta utilizar **três** aspas duplas para delimitar o texto, seguidas de uma quebra de linha. Por exemplo:

String mensagem = """

Olá, mundo!

Este é um Text Block.

Ele permite escrever textos com múltiplas linhas

sem precisar usar caracteres de escape ou quebras de linha manualmente ou concatenações.

""";

Nesse exemplo, a variável mensagem recebe um Text Block com o texto "Olá, mundo! Este é um Text Block. Ele permite escrever textos com múltiplas linhas sem precisar usar caracteres de escape ou quebras de linha manualmente!".