

Ao compreender que além dos tipos primitivos de dados, o Java oferece os tipos de referência (classes) correspondentes a todos os tipos primitivos, pesquise e identifique um método que lhes chame a atenção em cada classe dos tipos Wrapper: Integer, Double, Character, Boolean e String.

Ao final implemente um pequeno algoritmo que utilize pelo menos três dos métodos pesquisados.

<b>Integer</b>	<code>parseInt(String s)</code>	Essencial para <b>conversão de tipos</b> . Permite transformar uma <code>String</code> que representa um número (por exemplo, "123") diretamente em um valor primitivo <code>int</code> . É o método mais comum para processar entradas de usuário (que geralmente são lidas como <code>String</code> ) em operações matemáticas.
<b>Double</b>	<code>isNaN()</code>	Determina se o objeto <code>Double</code> representa o valor " <b>Not-a-Number</b> " (Não é um Número). É crucial para <b>validação de cálculos complexos</b> , pois evita que erros de divisão por zero ( <code>0.0 / 0.0</code> ) ou operações inválidas propaguem resultados numéricos incorretos pelo sistema.
<b>Character</b>	<code>isLetterOrDigit(char ch)</code>	Um método de <b>validação de caracteres</b> extremamente útil. Ele retorna <code>true</code> se o caractere for uma letra (de 'a' a 'z' ou 'A' a 'Z') ou um dígito (de '0' a '9'). É ótimo para verificar a validade de entradas de texto, senhas ou nomes de variáveis.
<b>Boolean</b>	<code>parseBoolean(String s)</code>	Semelhante ao <code>parseInt</code> , este método converte uma <code>String</code> para um valor primitivo <code>boolean</code> . Ele

## String

contains(CharSequence  
s)

retorna true se a *string* for "true" (ignorando maiúsculas/minúsculas) e false para qualquer outro valor. É muito prático para processar configurações lidas de arquivos de texto.

Essencial para a **manipulação de textos**. Retorna true se a *string* contém a sequência de caracteres especificada. É uma maneira simples e direta de verificar a presença de palavras-chave, frases ou substrings dentro de um texto maior.