

Bibliotecas: módulo java.base

Bibliotecas: módulo `java.base`

- ❖ Já vimos o conceito de pacotes em Java, basicamente eles existem para evitar conflitos de nomes e são organizados em diretórios que seguem a estrutura de um domínio.
- ❖ Mas o que é um **Módulo**.
- ❖ O conceito de módulo em java introduzido no Java 9 constitui, grosso modo, um conjunto de pacotes relacionados.
- ❖ Sendo assim o Java definiu o módulo **`java.base`** que contém os pacotes contendo as classes mais básicas da linguagem.

Bibliotecas: módulo java.base

- ❖ Dentre esses pacotes temos por exemplo:
 - ❖ **java.math** - Contém operações matemáticas em geral.
 - ❖ **java.net** - Classes para aplicações em rede.
 - ❖ **java.io** - Manipulação de arquivos.
 - ❖ **java.time** - para manipular hora, datas e medições de tempo.
 - ❖ **java.util** - contém classes das chamadas Java Collections (Array, List, Map, etc.) , para geração de números aleatórios, e muitos outros.
 - ❖ **javax.swing** - classes para a criação de interfaces gráficas.

Bibliotecas: módulo `java.base`

- ❖ Além dos pacotes mencionados o módulo **`java.base`** possui diversas outras, seu conteúdo extenso não nos permite analisá-los aqui.
- ❖ Portanto iremos focar em apenas alguns deles.
 - ❖ `java.math`, `java.util` (collections basicamente) e `java.io` e um pouco do `java.swing`.

java.io

- ❖ Consulte os detalhes das classes mencionadas em:

<https://docs.oracle.com/javase/7/docs/api/java/io/package-summary.html>

java.io: Leitura e Escrita de Arquivos

- ❖ Veremos como utilizar o **java.io** para ler e escrever dados em arquivos.

Classe	Descrição
java.io.FileWriter	Representa um arquivo aberto para leitura.
java.io.FileReader	Representa um arquivo aberto para escrita.
java.io.BufferedReader	Ler texto de um fluxo de caracteres de maneira eficiente para caracteres, vetores e linhas.
java.io.BufferedWriter	Escreve texto de um fluxo de caracteres de maneira eficiente para caracteres, vetores e linhas.

- ❖ Existem diversas forma de manipular um arquivo em Java, porém veremos está que é a mais básica utilizando BufferedReader and BufferedWriter, para mais informações vejam o pacote **java.nio**

Vamos ao um exemplo...

- ❖ Código no nosso repositórios... [link aqui](#).