

Disciplina Regular 1

Fundamentos do

Desenvolvimento Java

Graduação em Engenharia de Software - 2020

Etapa 4 Aula 2

Tratamento de Erros e **Processamento de Arquivos**

Competências Trabalhadas Nesta Etapa

- Utilizar classes padrão fundamentais do Java:
 - Manipular arquivos e diretórios com a classe File
 - Ler e gravar dados em arquivos utilizando as classes do pacote java.io.

No Moodle esse conteúdo se refere à etapa 7

Streams

- Streams são canais de comunicação entre um programa e uma fonte de dados - fluxos.
- As fontes de dados podem ser: arquivos, trechos de memória e conexões de rede.
- Em Java não importa qual seja a fonte: todo o tratamento de dados é igual.
- Fontes das quais lemos dados são fluxos de entrada e aquelas nas quais gravamos dados são fluxos de saída.

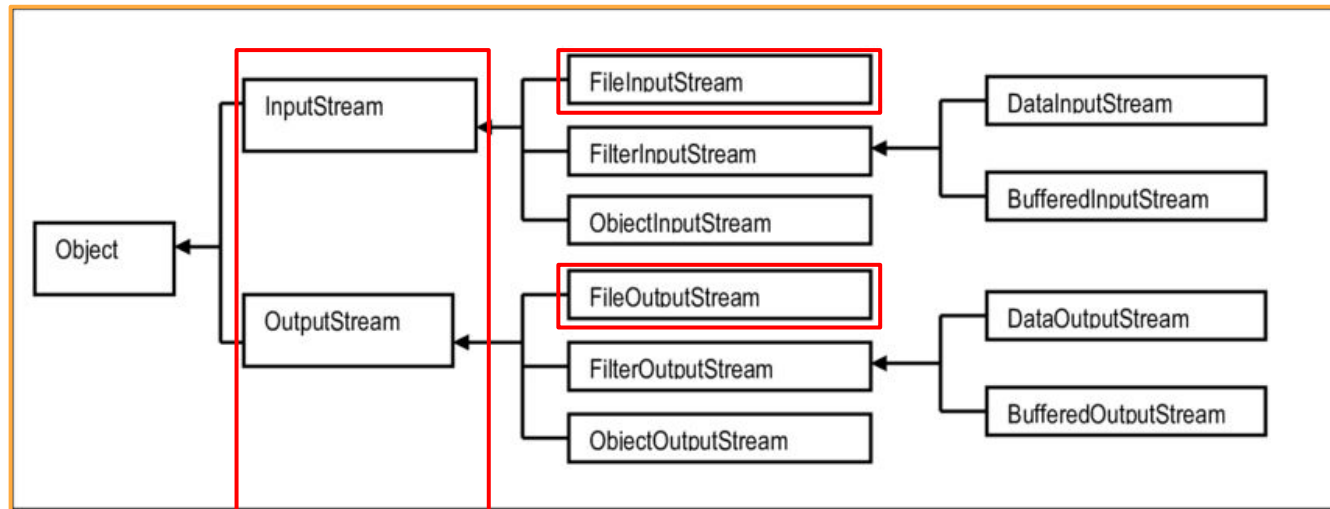
Streams

- Em Java, existem dezenas de classes que representam fluxos de entrada e saída.
- Estas classes são divididas em três grupos:
 - Classes Base Abstratas, subdivididas em classes de acesso binário e texto.
 - Classes de Acesso Físico.
 - Classes de Acesso Lógico.
- Todas as classes de *stream* estão definidas no pacote java.io.

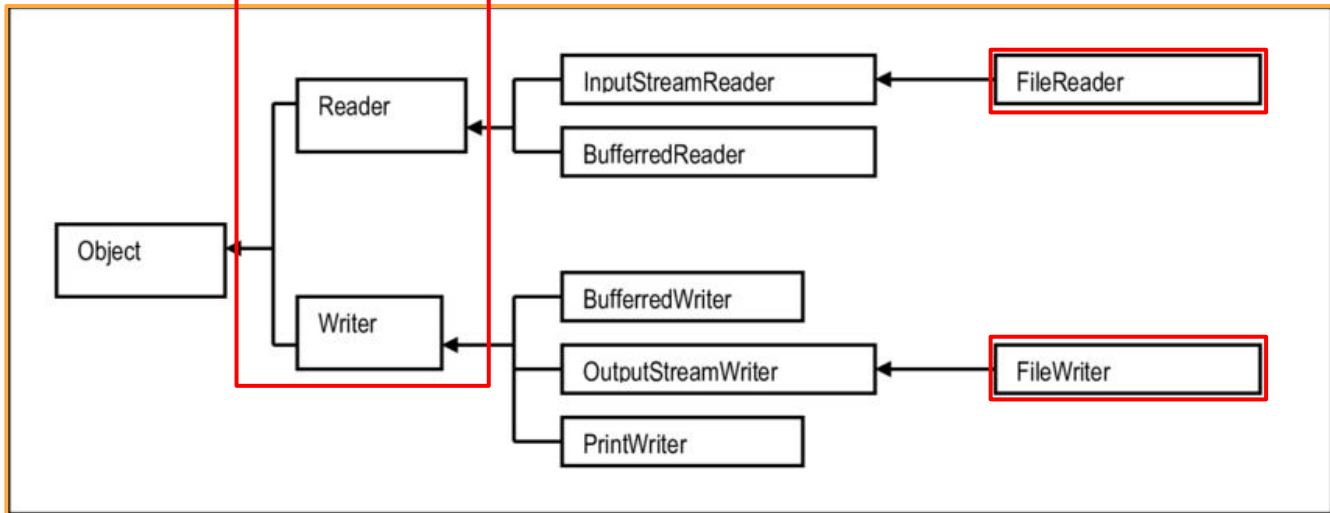
Streams

- As classes base abstratas definem um padrão de acesso(E/S) a dados.
- Podem acessar bytes ou caracteres.
- Acesso a bytes:
 - InputStream, leitura de dados.
 - OutputStream, gravação de dados.
- Acesso a texto (formato UNICODE):
 - Reader, leitura de dados.
 - Writer, gravação de dados.

**Classes
de
acesso
binário
byte**



**Classes
de
acesso
texto
char**



Classes de Arquivo

- As classes de acesso físico definem formas de manipulação de dados diretamente da fonte de dados.
- A leitura de dados de um arquivo pode ser feita com as classes **FileInputStream** ou **FileReader**.
- Para gravar dados em um arquivo usamos **FileOutputStream** e **FileWriter**.

Classes de Arquivo

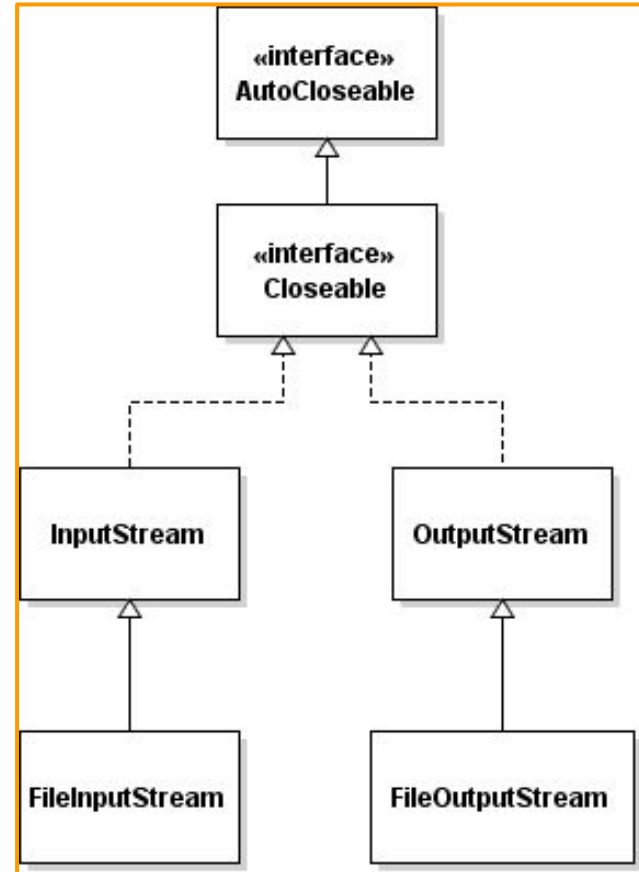
- Ao criar um objeto de qualquer uma destas classes, será aberto o arquivo passado como parâmetro.
- Para ler dados, temos o comando **read()**.
- Para gravar dados, usamos o **write()**.
- Para fechar um arquivo, usamos o **close()** - **você verá que este método não será necessário.**
- Estes comandos podem gerar exceções derivadas de `IOException`.

Classe File

- Usamos a classe File para:
 - Representar um arquivo ou diretório.
 - Criar um arquivo ou diretório.
 - Verificar os atributos de um arquivo ou diretório.
 - Renomear um arquivo ou diretório.
 - Remover um arquivo ou diretório.
 - Recuperar informações diversas sobre um arquivo ou diretório.

try-with-resources

- try-with-resources é uma instrução try que declara um ou mais recursos.
- A instrução try-with-resources garante que cada recurso seja fechado no final da instrução.
- Qualquer objeto que implementa `java.lang.AutoCloseable`, que inclui todos os objetos que implementam `java.io.Closeable`, pode ser usado como um recurso.
- `FileInputStream` e `FileOutputStream` podem usar essa instrução.



Exemplo

```
try (FileInputStream fisEstados = new FileInputStream("estados.csv");
    FileInputStream fisMunicipios = new FileInputStream("municipios.csv")) {

    //-----
    byte[] bytes = new byte[fisEstados.available()];
    fisEstados.read(bytes);
    String estados = new String(bytes);
    //-----
    bytes = new byte[fisMunicipios.available()];
    fisMunicipios.read(bytes);
    String municipios = new String(bytes);
    //-----

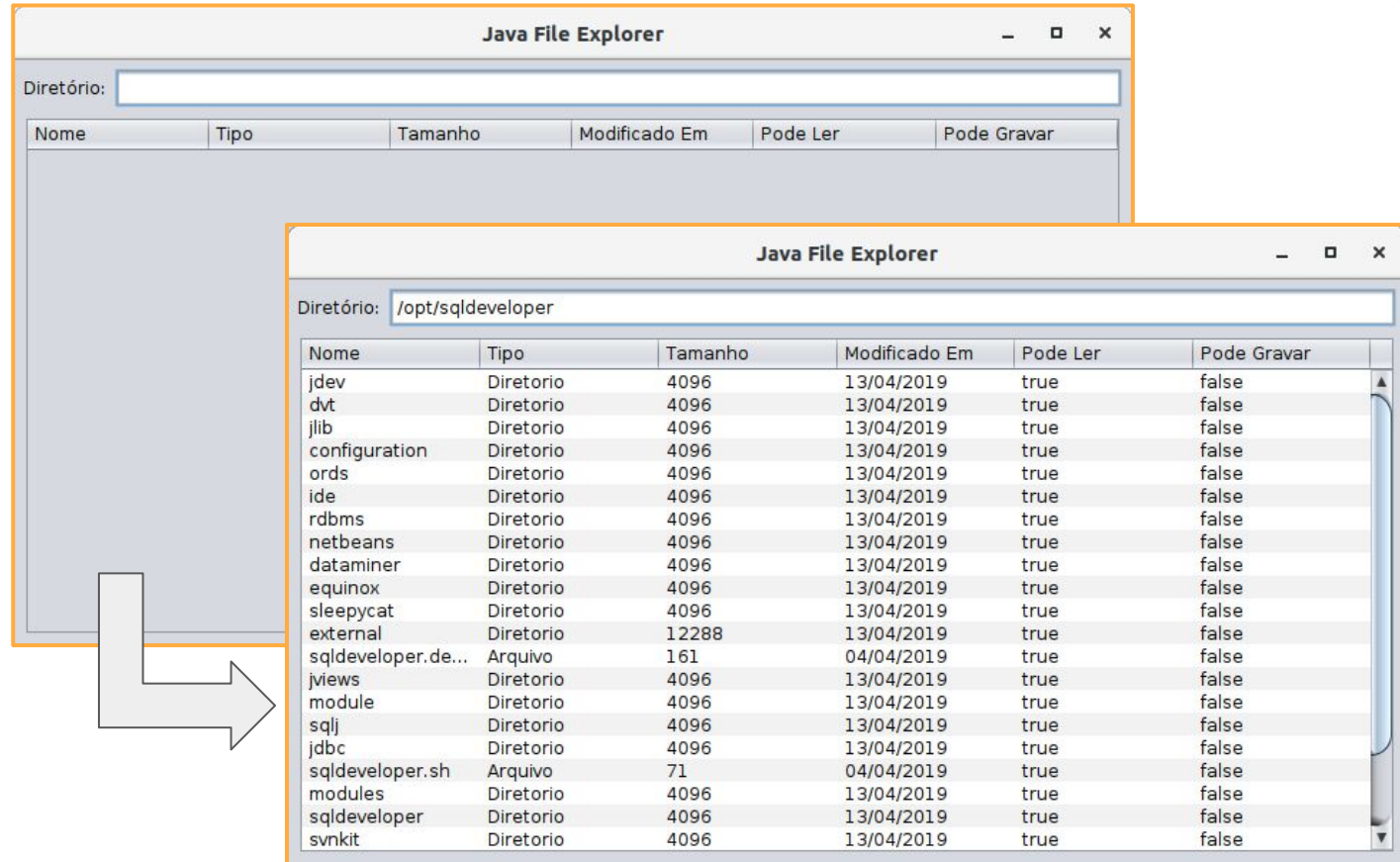
} catch (FileNotFoundException ex) {

    JOptionPane.showMessageDialog(this,
        "ERRO: Arquivo não encontrado",
        "ERRO",
        JOptionPane.ERROR_MESSAGE);
} catch (IOException ex) {

    JOptionPane.showMessageDialog(this,
        "ERRO: Não é possível ler o arquivo",
        "ERRO",
        JOptionPane.ERROR_MESSAGE);
}
```

Exercício

Construir uma aplicação que dado um diretório liste os arquivos com os seus dados, conforme as figuras:



Exercício

- Criar um programa para ler os arquivos `municipios.csv` e `estados.csv` exibindo os dados como ao lado.
 - Ler o arquivo para a memória.
 - Separar as linhas (terminadas por `\n`).
 - Separar os campos.



The screenshot shows a window titled "Estados e Municípios do Brasil" with a table containing the following data:

Sigla	Estado	Município	Latitude	Longitude
RO	Rondônia	Alta Floresta D'Oeste	-11.9283	-61.9953
		Alto Alegre dos Países	-12.132	-61.835
		Alto Paraíso	-9.71429	-63.3188
		Alvorada D'Oeste	-11.3463	-62.2847
		Ariquemes	-9.90571	-63.0325
		Buritis	-10.1943	-63.8324
		Cabixi	-13.4945	-60.552
		Cacaulândia	-10.349	-62.9043
		Cacoal	-11.4343	-61.4562
		Campo Novo de Rondonia	-10.5712	-63.6266
		Candeias do Jamari	-8.7907	-63.7005
		Castanheiras	-11.4253	-61.9482
		Cerejeiras	-13.187	-60.8168
		Chupinguaia	-12.5611	-60.8877
		Colorado do Oeste	-13.1174	-60.5454
		Corumbiara	-12.9551	-60.8947
		Costa Marques	-12.4367	-64.228
		Cujubim	-9.36065	-62.5846
		Espinaço D'Oeste	-11.5266	-61.0252

At the bottom of the window is a button labeled "Carregar Dados".

Exercício

Construir uma aplicação simples que permita ler, editar e gravar arquivos-texto.

