



Prof. Robson L. F. Cordeiro

Lista 9 – Streams e arquivos

1. Escreva um programa que recebe o caminho de um diretório e lista os arquivos encontrados no próprio diretório, e nos subdiretórios recursivamente.
2. Reescreva o programa do Exercício 1, fazendo com que seu programa mostre apenas os arquivos que são de um tipo específico a sua escolha, por exemplo, arquivos executáveis (*.exe) – pesquise e use a interface `FilenameFilter`.
3. A classe `LineNumberReader` é uma classe da API Java que lê dados de um arquivo texto mantendo um contador que indica em qual linha do arquivo está ocorrendo a leitura.

O construtor do `LineNumberReader` é "`LineNumberReader(Reader in)`"; seu método de leitura de linha é "`String readLine()`" que retorna `null` quando o stream chegou ao fim; e seu método para saber o número da linha sendo lida é "`int getLineNumber()`".

Escreva um programa que leia um arquivo texto e que indique quando a linha 10 tiver sido lida, ou que informe que o arquivo não possui 10 linhas. Use a classe `FileReader` (extends `Reader`) para construir um `LineNumberReader`.

4. A classe `ByteArrayInputStream` é capaz de ler bytes de um array, isto é, trata-se de uma maneira de ler dados da memória por meio de um fluxo (stream) de dados. O construtor do `ByteArrayInputStream` é "`ByteArrayInputStream(byte array[])`";

Escreva um programa que cria um byte array contendo bytes correspondentes aos caracteres de seu nome (use o método `getBytes` da classe `String`). Em seguida use um `FileOutputStream` para escrever o array de bytes em um arquivo.

Para entrega: códigos dos projetos NetBeans referentes aos exercícios acima em um arquivo zip → entregar via Tidia→Atividades

5. Escreva um programa que escreve e lê do disco objetos de uma classe `DadosPessoais`, a qual possui um atributo que é do tipo `Endereco`, i.e., uma outra classe, dentre outros atributos à sua escolha. Escreva as classes correspondentes e o programa usando as classes `GZIPOutputStream`, `ObjectOutputStream`, `GZIPInputStream` e `ObjectInputStream`.
6. Escreva um programa que lê um arquivo texto e toda vez que encontra uma string "muito" a substitui por uma string "pouco".
7. Escreva uma classe denominada `MeuLineNumberReader` que possua um método `readLine`, o qual deverá retornar uma linha iniciada pelo número da linha lida seguida de ":".
8. Escreva um programa que recebe um caminho de diretório e que, apenas para os arquivos.mp3 encontrados, renomeie os arquivos da seguinte maneira:
 - se houver algarismos ou hifens no nome do arquivo, eles serão removidos;
 - ordene os arquivos encontrados por tamanho dos arquivos;
 - coloque no início do arquivo um contador formado por quatro algarismos de acordo com a ordenação.

Exemplo: `musica_do_ano278.mp3` → `0389-musica_do_ano.mp3`

→ para ordenação pesquise e use a interface `Comparator` - http://www.tutorialspoint.com/java/java_using_comparator.htm