



# **Formação Analista Desenvolvedor Java**

## **Exercícios Propostos**

**Fundamentos de I/O**

# 1 Exercício

Escreva uma aplicação para gerenciar itens de uma lista de compras. Ela deve funcionar da seguinte forma.

Quando executada, a aplicação deve solicitar que o usuário escreva o nome de um item e pressione *ENTER*. Quando fizer isto, este item deve ser armazenado num arquivo `lista.txt`. A aplicação fica em loop solicitando um item após o outro, até o momento que o usuário digitar o item *"0"*. Quando ele fizer isto, a aplicação termina. Outro detalhe é que não deve ser permitido que o usuário insira itens vazios (como, por exemplo, só espaços em branco ou apenas um *ENTER*). Nestes casos, a aplicação deve desconsiderar o item e solicitar o próximo.

Mais um requisito da aplicação é que, quando aberta, ela deve verificar se o arquivo `lista.txt` existe e se possui itens cadastrados. Caso isto seja verdadeiro, antes de solicitar a entrada de novos itens, os itens já cadastrados devem ser mostrados na tela. Quando o usuário adicionar novos itens, os itens já cadastrados não devem ser apagados (os novos devem ser adicionados no fim do arquivo).

**Dica:** Para ler os dados do teclado, experimente utilizar o `Scanner` (lembre-se de que o `System.in` é a *input stream* padrão que lê os dados do teclado). Para ler e escrever em arquivos, `FileReader` e `FileWriter` podem facilitar o processo. A classe `PrintWriter` pode ser utilizada em conjunto com o `FileWriter` para facilitar ainda mais a escrita de strings diretamente no arquivo com quebras de linha no final.

# 2 Exercício

Crie um aplicativo capaz de gravar os dados de objetos em um arquivo em forma de bytes e também de lê-los.

Primeiramente crie duas classes, `Livro` e `Autor`, de acordo com o demonstrado no diagrama abaixo:



Crie também uma interface `Recordable`, que declara dois métodos:

```
public void read(DataInputStream in) throws IOException
public void write(DataOutputStream out) throws IOException
```

Faça com que `Livro` e `Autor` implementem esta interface. Classes que implementam esta interface devem codificar a forma como seus próprios atributos são gravados e lidos no arquivo, através de chamadas aos objetos `in` e `out`.

A aplicação deve pedir para que o usuário escolha entre duas opções: gravar ou ler os dados. Caso a opção de gravar seja escolhida, dois objetos `Livro` devem ser criados e gravados em um arquivo. Já se a opção de leitura for escolhida, estes dois objetos devem ser lidos do arquivo e impressos no console.

**Dica:** Objetos `DataInputStream` e `DataOutputStream` possuem métodos para ler/gravar diversos tipos de dados, como `String`, `int`, `double`, etc. Eles encapsulam uma `InputStream` e uma `OutputStream`, respectivamente. Tenha em mente que a ordem de leitura dos dados do arquivo deve ser a mesma ordem de escrita.