



PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS

Lista 05

Professora: *Marta Noronha*

Disciplina: *Algoritmos e Estrutura de Dados II*
Data de entrega: 22/09/2024

Requisitos

1. Todos os programas deverão ser desenvolvidos na linguagem de programação Java.
2. Cópias, se existirem, serão encaminhadas ao colegiado de coordenação didática do curso.
3. Fique atento ao charset dos arquivos de entrada e saída. Recomenda-se a utilização dos métodos da classe `MyIO.java` para leitura de dados do teclado. É necessário definir o charset a ser utilizado antes de começar a leitura de dados do teclado, da seguinte forma:
`MyIO.setCharset("UTF-8").`
4. As saídas esperadas, cadastradas no VERDE pelo professor, foram geradas empregando-se:
`MyIO.println().`
5. Em cada submissão, enviar apenas um arquivo (.java). A regra será necessária para a submissão de exercícios no VERDE e no identificador de plágios utilizado na disciplina.
6. A resolução (código) de cada exercício deverá ser submetida ao VERDE.
7. A execução do código submetido será realizada automaticamente pelo VERDE, mas o código será analisado e validado pelo professor.
8. Se for necessário ler o arquivo "*pub.in*", o mesmo será disponibilizado para *download* no VERDE juntamente com o arquivo "*pub.out*".

Exercícios - Criação da classe Livros

Crie uma classe Livro com os seguintes atributos privados:

- ISBN (Long);
- titulo (String);
- autor principal (String);
- segundo autor (String);
- categoria (String);
- descricao (String);
- ano publicação (int);
- quantidade páginas (int);
- nota avaliação (double / float);
- quantidade avaliações (int);

A classe também deve conter, obrigatoriamente, ao menos, dois construtores (1. *padrão (default)*; 2. *título/autor/ano publicação*), e os métodos *gets*, *sets*, métodos *clone()*, *ler()*, *imprimir()* e *toString()*.

O método *clone()* deve retornar um objeto da mesma classe e contendo os mesmos valores de atributos do atual objeto analisado (**vide exemplo do método clone no slide "Unidade 0 - Nivelamento - Ponteiros (C/C++) e Referências (Java)"** postado na disciplina teórica). **Não é permitido usar a interface Cloneable para esta finalidade.**

O método *ler()* deve receber cada linha do arquivo como parâmetro e armazenar os valores contidos em cada linha nos atributos de cada objeto que foi instanciado.

O método *imprimir()* exibe os valores dos atributos do objeto Livro, conforme o modelo indicado no fim deste documento, conforme mostrado AQUI.

O método *toString()* deve ser criado para permitir a impressão da classe, sem necessidade de invocação do método *imprimir()*.

Após a criação da classe, deve ser criado um mecanismo para processamento de uma entrada de dados. A entrada de dados é dividida em 2 partes:

- Parte 1: Armazenamento de informações em vetor;
- Parte 2: Pesquisa de informações armazenadas no vetor criado na parte 1.

Parte 1 (Leitura de Arquivo): Armazenamento de informações contidos no arquivo livros.txt em vetor

Seu programa deve ler um arquivo-texto chamado "**livros.txt**" que, no VERDE, localiza-se na pasta **"/tmp"**.

O aluno(a) deve preencher um vetor de objetos da classe Livro com os dados dos diversos livros informados no arquivo "livros.txt". Use um `ArrayList<Livros>` para aumentar dinamicamente o *array* contendo os livros que serão incluídos. Consulte um guia disponível em Java `ArrayList` para compreender o funcionamento do *ArrayList*.

Cada uma das linhas seguintes apresenta os dados de um livro, separados pelo símbolo "|". Os dados possuem, em ordem, as seguintes informações:

- ISBN (long);
- titulo (String);
- autor principal (String);
- segundo autor (String);
- categoria (String);
- descricao (String);
- ano publicação (int);
- quantidade páginas (int);
- nota avaliação (double / float);
- quantidade avaliações (int);

Importante: O caractere "|" é utilizado para expressões regulares. Para dividir uma palavra por "|", utilize a seguinte sequência: "\\|".

A última linha do arquivo "**livros.txt**" é vazia.

Parte 2: Pesquisa de informações armazenadas no vetor (ArrayList) criado na parte 1.

Após o processamento da primeira parte da entrada de dados, o programa deve processar a entrada contida no arquivo *pub.in*. Cada linha da segunda parte contém, em ordem, as seguintes informações:

- Título;
- Ano de lançamento;
- Autor principal.

A última linha do arquivo *pub.in* contém a palavra FIM.

As entradas devem ser pesquisadas no vetor de livros.

Para cada livro encontrado, escreva na saída padrão uma linha contendo as informações bibliográficas do livro, com os dados do registro correspondente. **Somente a primeira ocorrência de um livro deve ser apresentada na saída.** A saída padrão deve obedecer o seguinte formato (*não incluir as chaves*):

{autor principal}, {segundo autor}. {titulo}. {ano publicação}. ISBN: {isbn}.

Exemplos:

Rafael C. Gonzalez, Richard E. Woods. Digital Image Processing. 2018. ISBN: 9789353062989.

Nívio Ziviani. Projeto De Algoritmos Com Implementações Em Pascal e C. 2010. ISBN: 9788522110506.

O programa deve contabilizar a quantidade de livros lidos na entrada padrão, neste caso *pub.in*, que não foram encontrados no arquivo *livros.txt*. Ao final do arquivo, deverá ser impresso a informação:

Quantidade de livros não encontrados: {total de livros}

(Obs: as chaves não devem ser impressas)