



PUC Minas

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS

Lista de Exercícios 04

Professora: *Marta Noronha*

Disciplina: *Laboratório de Algoritmos e Estrutura de Dados II*
Data de entrega: *08/09/2024*

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("ISO-8859-1")`.
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.

Base de Dados

Para construção desta atividade foi utilizada a base de dados 7k Books.

Este conjunto de dados contém uma lista simplificada de livros populares, incluindo informações de ISBN13, ISBN10, título, subtítulo, categoria, descrição, ano de publicação, número de páginas, avaliação (nota) média e quantidade de avaliações. O data set foi baseado na base de dados Goodread books.

Durante a etapa de filtragem, foram selecionados os livros mais populares, usando como critério a quantidade de avaliações. Foram removidos registros com informações incompletas ou que apresentavam possíveis conteúdos sensíveis / inadequados à disciplina.

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 vetores: um aleatório e outro ordenado;
- Parte 2: Pesquisa de informações armazenadas nos vetores por pesquisa sequencial e pesquisa binária.

Parte 1: Armazenamento de informações em vetores: um aleatório e outro ordenado

A classe ArrayList NÃO poderá ser usada para esta atividade.

O aluno(a) deve preencher dois vetores de objetos da classe Livro com os dados dos diversos livros informados na entrada padrão (teclado - *pub.in*). A primeira linha contém a quantidade de livros que devem ser armazenados em cada vetor de livros, ou seja, os vetores possuem o mesmo tamanho.

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: "\\|".

Na primeira parte do *pub.in*, a qual é encerrada com a palavra FIM, estão os dados dos livros que devem ser armazenados no primeiro vetor. Estes livros não estão ordenados.

Na segunda parte do *pub.in*, a qual é encerrada com a palavra FIM, estão os dados dos livros que devem ser armazenados no segundo vetor. Estes livros estão ordenados por título.

Parte 2: Pesquisa de informações armazenadas nos vetores por pesquisa sequencial e pesquisa binária.

Antes de realizar as pesquisas nos vetores:

- Implemente no programa principal (*main*) um procedimento para a pesquisa sequencial que receba o vetor de livros não ordenados e a string contendo o nome do livro a ser buscado. Caso existam mais de um livro com o mesmo título (por exemplo, um livro que foi atualizado), somente deve ser impressa a informação do primeiro livro encontrado, conforme o exemplo apresentado AQUI.
- Implemente no programa principal (*main*) um procedimento para a pesquisa binária que receba o vetor de livros não ordenados e a string contendo o nome do livro a ser buscado. Caso existam mais de um livro com o mesmo título (por exemplo, um livro que foi atualizado), somente deve ser impressa a informação do primeiro livro encontrado, conforme o exemplo apresentado AQUI.
- Ambos procedimentos devem contabilizar o total de comparações realizados. *Obs: Modifique o algoritmo de pesquisa binária, visto em sala, para que a comparação entre o livro no vetor e o livro buscado via string seja realizada somente uma única vez.*

Após o armazenamento dos dados nos dois vetores, o programa deve processar a terceira parte da entrada padrão (a qual também termina com a palavra FIM). Cada linha da terceira parte contém somente o título do livro que deverá ser buscado em ambos os vetores de livros.

Para cada linha lida, use a pesquisa sequencial para contabilizar a quantidade de comparações, imprimindo a saída padrão. Em seguida, pesquise o livro contido nessa linha pela pesquisa binária. Novamente contabilize as comparações e imprima a saída padrão solicitada.

A saída padrão deve obedecer o seguinte formato (*não incluir as chaves*):

{autor principal}, {segundo autor}. {título}. {ano publicação}. ISBN: {isbn}. Sequencial: {quantidade de comparações na pesquisa sequencial}, binaria: {quantidade de comparações na pesquisa binária}

Exemplos:

Edward Gorey. The Gashlycrumb Tinies, Or, After the Outing. 1998. ISBN: 9780747541608. Sequencial: 33, binaria: 10

Graham Greene, Robert Stone. The quiet American. 1955. ISBN: 9780143039020. Sequencial: 594, binaria: 9

Sabendo que a pesquisa sequencial é $\Theta(n)$ e a pesquisa binária é $\Theta(\log(n))$, compare os resultados da complexidade apontada na teoria com a prática.