



PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS

Lista de Exercícios 08

Professor: *Marta Noronha*

Disciplina: *Laboratório de Algoritmos e Estrutura de Dados II*
Data de entrega: *18/10/2024*

Requisitos

1. Todos os programas deverão ser desenvolvidos na linguagem de programação Java.
2. Essas práticas poderão ser desenvolvidas em grupos de, no máximo, dois integrantes.
3. Cópias, se existirem, serão encaminhadas ao colegiado de coordenação didática do curso.
4. 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")`.
5. As saídas esperadas, cadastradas no VERDE pelo professor, foram geradas empregando-se:
`System.out.println()`.
6. 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.
7. A resolução (código) de cada exercício deverá ser submetida ao VERDE.
8. 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.

OBS: As filas estática e dinâmica serão avaliadas em 1 ponto cada.

Exercício 1 - Fila circular implementada por vetor

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 no vetor criado na Parte 1.

Após a leitura do arquivo, na primeira parte do trabalho, o programa deve ler informações da entrada padrão (*pub.in*). Cada linha da segunda parte contém, em ordem, as seguintes informações:

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

As entradas devem ser pesquisadas no vetor de livros. Os registros encontrados na pesquisa devem ser armazenados em uma **fila circular, capaz de armazenar até 100 livros simultaneamente**, que será utilizado na Parte 3 desta prática.

Os métodos de sua fila circular devem operar conforme descrito a seguir, respeitando-se parâmetros e tipos de retorno:

- Sua classe Fila deverá ter dois construtores;
- `void enfileirar (Livro livro):` enfileira um objeto do tipo Livro;
- `Livro desenfileirar ():` desenfileira e retorna o livro da frente da fila;
- `void mostrar ():` para todos os objetos do tipo Livro presentes na fila, exibe a posição do objeto na fila seguida dos valores de seus atributos (observe o formato de cada linha da saída esperada);
- `double obterSomaDaNotaMedia ():` calcula a soma de todas as notas médias da avaliação de todos os livros presentes na lista.

A execução desta parte do trabalho deve ser finalizada após a leitura da palavra FIM.

Parte 3: Enfileiramentos e desenfileiramentos na fila criada na Parte 2.

A primeira linha da segunda parte da entrada padrão apresenta um número inteiro n indicando a quantidade de livros que serão enfileirados ou desenfileirados. Nas próximas n linhas, tem-se n comandos de enfileiramento ou desenfileiramento, que devem ser processados. Cada uma dessas linhas tem uma palavra de comando, conforme descrito a seguir:

- E: enfileirar;
- D: desenfileirar.

No caso dos comandos de enfileiramento, temos também uma string contendo as seguintes informações, que devem ser pesquisadas no vetor de livros da Parte 1:

- Título;
- Ano de lançamento;
- Nota de avaliação.

Após encontrar o registro no vetor de livros, adicioná-lo à fila. **Se, no momento de execução da operação enfileirar, a fila estiver cheia, antes de enfileirar um livro será necessário desenfileirar outro.**

A saída padrão será composta por:

- um número inteiro correspondente à média da avaliação de todos os livros contidos na fila, após cada enfileiramento.
- Uma linha para cada livro desenfileirado, sendo que essa informação será constituída pela string “(D)” seguida dos atributos desse livro.
- Ao final da execução, devem ser impressas as informações dos livros armazenados na fila (observe o formato de cada linha da saída esperada).

Para o caso do desenfileiramento deve ser impresso:

(D) [categoria] [nota] [qtd. avaliação] {autor principal}, {segundo autor}. {titulo}. {ano publicação}. ISBN: {isbn}.

Exemplo da saída após desenfileirar:

(D) [Fiction] [4.16] [59865] Isaac Asimov. *The Caves of Steel*. 1993. ISBN: 9780586008355.

Para o caso da impressão dos livros que estão na fila, deve ser impresso:

[posição na fila] [categoria] [nota] [qtd. avaliação] {autor principal}, {segundo autor}. {titulo}. {ano publicação}. ISBN: {isbn}.

Exemplo da saída:

[0][Fiction] [4.16] [59865] Isaac Asimov. *The Caves of Steel*. 1993. ISBN: 9780586008355.

Exercício 02 - Fila implementada com alocação dinâmica

Refaça o exercício anterior usando alocação dinâmica de memória.

Neste exercício, sua classe Fila deverá ter apenas um construtor. Portanto não deve conter um limite máximo de registros como na fila estática.