

Lista 02

Professora: *Marta Noronha*

Disciplina: *Algoritmos e Estrutura de Dados II*

Data de entrega: 09/03/2024

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:
`System.out.println()`.
5. Os arquivos `pub.in` e `pub.out` estarão disponíveis no VERDE e forem necessários para a execução dessa atividade.
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.

Criação da classe Jogos

O conjunto de dados utilizado neste exercício foi disponibilizado em Kaggle.

Crie uma classe `Jogo` com os seguintes atributos declarados com modo de acesso privado:

- `rank` (int): classificação (posição) do jogo em relação aos demais.
- `nome do jogo` (String): o nome do jogo.
- `plataforma` (String): Onde o jogo pode ser executado (Wii, DS, entre outros).
- `ano` (int): ano de lançamento.
- `gênero` (String): esportes, corrida, entre outros.
- `editora` (String): empresa que disponibiliza o jogo (produtora).
- `NA_Vendas` (double): Vendas do jogo na América do Norte.
- `EU_Vendas` (double): Vendas do jogo na Europa.
- `JP_Vendas` (double): Vendas do jogo no Japão.

- Outras_Vendas (double): Vendas do jogo em outros lugares.
- Vendas_Global (double): Vendas global do jogo.

A classe também deve conter, obrigatoriamente, ao menos, dois construtores (1. padrão (default); 2. (nome do jogo)/plataforma/ano), 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 Jogo, 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: Armazenamento de informações em vetor

O aluno(a) deve preencher um vetor de objetos da classe Jogo com os dados dos diversos jogos informados na entrada padrão (teclado - *pub.in*).

A primeira linha contém a quantidade de jogos que devem ser armazenados em um vetor.

Cada uma das linhas seguintes apresenta os dados de um jogo, separados pelo símbolo “;”. Os dados possuem, em ordem, as seguintes informações:

- rank (int);
- nome do jogo (String);
- plataforma (String);
- ano (int);
- gênero (String);
- editora (String);
- NA_Vendas (double);
- EU_Vendas (double);
- JP_Vendas (double);
- Outras_Vendas (double);
- Vendas_Global (double);

A última linha dessa primeira parte da entrada padrão contém apenas a palavra FIM.

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

Após o processamento da primeira parte da entrada de dados, o programa deve processar a segunda parte da entrada padrão. Cada linha da segunda parte contém, em ordem, as seguintes informações:

- nome do jogo;
- ano;
- editora.

As entradas devem ser pesquisadas no vetor de jogos.

Para cada jogo encontrado, escreva na saída padrão uma linha contendo as informações do jogo, com os dados do registro correspondente. Caso um jogo apareça no arquivo *pub.in* mais de uma vez, somente a primeira ocorrência de um jogo deve ser apresentada na saída.

A saída padrão deve obedecer o seguinte formato (não incluir as chaves): {nome do jogo}. {editora}. {ano}.
Vendas global: {Vendas_Global}.

Exemplo:

Sega. Puyo Puyo Tetris. 2014. Vendas global: 0.05.