

# PROGRAMAÇÃO II – TRABALHO 01

## 1. Objetivo

Construir uma aplicação que mostre algumas informações sobre o acervo de uma biblioteca. Para isso, implemente uma classe chamada *Livro* que deve conter os seguintes atributos:

- Código do livro (um número inteiro)
- Título do livro
- Um vetor contendo objetos de uma classe *Autor*
- Ano de publicação
- Nome da editora do livro
- Um vetor contendo objetos *String* com a listagem de assuntos tratados no livro (por exemplo, GEOGRAFIA, HISTÓRIA, JAVA, MATEMÁTICA, etc.)

Faça um método para essa classe que retorne um *string* contendo a referência bibliográfica do livro no seguinte formato:

SOBRENOME1, Nome1; SOBRENOME2, Nome2; ... . **Título do Livro**. Editora, Ano de publicação.

Lembrando que o sobrenome de cada autor deve ser exibido em letras maiúsculas. As reticências (...) acima representam a continuidade da lista de autores do livro (ou seja, um livro NÃO terá exatamente dois autores, podendo possuir menos ou mais).

Crie uma classe chamada *Autor* que conterá informações sobre autores. Essa classe deve conter os seguintes atributos:

- Sobrenome do autor
- Primeiro nome do autor.

Crie um construtor para essa classe que permita se inicializar os atributos da mesma.

Para ambas as classes, crie construtores que julgar necessário e também outros métodos que achar conveniente.

Crie uma terceira classe chamada *ListagemDeLivros* que deverá conter apenas um atributo: um vetor de objetos da classe *Livro*. O código base desta classe está disponibilizado junto com o PDF deste trabalho no Moodle.

Note que um construtor já foi adicionado à classe. Esse construtor inicializa o vetor de livros com alguns dados (você pode modificá-los, se desejar). A classe também já contém um método chamado *consultaPorTitulo* que recebe um *String* contendo o título ou parte do título de um livro e retorna um vetor de livros que contenham em seu título a *String* informada. Neste método note o uso da classe *ArrayList*. Esta classe permite-nos criar vetores dinâmicos (que podem crescer). Esta classe será útil nos demais métodos que deverão ser incluídos na classe.

Adiciona a essa classe mais três métodos:

- Um método que permita realizar uma consulta de livros por código. Esse método deve receber um número inteiro como parâmetro contendo o código do livro a ser pesquisado. O método deve retornar um único objeto da classe *Livro*, que contenha o código igual ao informado no parâmetro. Caso o código informado no parâmetro não seja igual ao de nenhum livro, o método deve retornar o valor *null*.
- Um método que permita realizar uma consulta por assunto. Esse método deve receber um *String* como parâmetro contendo o assunto desejado e deve retornar um vetor de objetos da classe *Livro* contendo os livros que tenham o assunto informado no parâmetro. DICA: use a classe *ArrayList* neste método nos moldes do que foi feito na consulta por título.
- Um método que permita realizar uma consulta por autor. Esse método deve receber um *String* como parâmetro contendo o sobrenome de um autor ou parte dele. O método deve retornar um vetor de objetos da classe *Livro* contendo os livros que tenham algum autor

com o sobrenome informado. DICA: use a classe *ArrayList* neste método nos moldes do que foi feito na consulta por título.

Crie um programa que utilize as classes anteriores e permita ao usuário consultar livros. Nesse programa o usuário deve escolher se deseja consultar livros por código, título, assunto ou nome do autor.

Na consulta por código, o programa deve pedir ao usuário que digite o código de um livro e deve mostrar os dados do livro encontrado, incluindo a string com a referência bibliográfica. Caso nenhum livro seja encontrado, o programa deve mostrar a mensagem "Nenhum livro encontrado".

Na consulta por título, o programa deve pedir ao usuário que digite o título de um livro (ou parte dele) e deve mostrar os códigos e títulos dos livros encontrados. Caso nenhum livro seja encontrado, o programa deve mostrar a mensagem "Nenhum livro encontrado".

Na consulta por assunto, o programa deve pedir ao usuário que digite o assunto desejado e deve mostrar os códigos e títulos dos livros encontrados. Caso nenhum livro seja encontrado, o programa deve mostrar a mensagem "Nenhum livro encontrado".

Na consulta por autor, o programa deve pedir ao usuário que digite o sobrenome do autor de um livro (ou parte dele) e deve mostrar os códigos e títulos dos livros encontrados. Caso nenhum livro seja encontrado, o programa deve mostrar a mensagem "Nenhum livro encontrado".

Utilize os métodos desenvolvidos na classe *ListagemDeLivros* na implementação das consultas. Por exemplo, na implementação da consulta de livros por título, basta solicitar ao usuário o String desejado, chamar o método *consultaPorTitulo* passando o String informado e mostrar os dados dos livros retornados pelo método.

O programa só deve terminar quando o usuário escolher sair (forneça essa opção a ele).

## 2. Entrega

Os códigos-fonte Java ou o projeto Netbeans devem ser entregues até a aula do dia 13/05.

## 3. Avaliação

O programa será avaliado levando em conta os seguintes aspectos:

- **FUNCIONAMENTO:** O programa precisa estar funcionando segundo os requisitos exigidos no trabalho.
- **USO DE OBJETOS:** O uso correto dos objetos, como visto em aula, será avaliado.
- **LEGIBILIDADE:** A clareza do código será avaliada, incluindo: nomes de variáveis adequados, correto uso de espaçamento e indentação do código, correto uso de comentários no código.
- **USO CORRETO DA IMPLEMENTAÇÃO DE CLASSES**
- **DEFESA:** Durante a aula do dia 13/05, será realizada uma defesa do trabalho, onde o aluno deverá demonstrar conhecimento sobre o código produzido e também conhecimento sobre o conteúdo relacionado ao trabalho (linguagem Java, conceitos de orientação a objetos, etc.).

O trabalho é INDIVIDUAL, e qualquer identificação de cópias de trabalhos será punida com a nota ZERO.