

MC322 – Programação Orientada a Objetos

Laboratório 03 – 2s2023

Prof. Dr. Bruno Cafeo (Professor)

Fillipe dos Santos Silva

Jesamin Zevallos Quispe

Vinicius Leme Soares

Discord: <https://discord.gg/mHcUFRDGgQ>

1 Introdução

A Biblioteca Universitária Multimídia é um ambiente dinâmico e repleto de atividades, onde diferentes tipos de itens multimídia, como livros físicos, livros eletrônicos, CDs e DVDs estão disponíveis para os membros. Estes membros incluem estudantes de graduação e pós-graduação, professores e funcionários. Para administrar as diversas operações da biblioteca, como empréstimos, renovações e reservas, o sistema necessita de um mecanismo eficiente de gerenciamento. Dessa forma, considerem o seguinte contexto:

- **Item Multimídia:** Representa os itens genéricos da Biblioteca, como Livro Físico, Livro Eletrônico, CD de Áudio e DVD de Vídeo.
- **Membro:** Representa os usuários da Biblioteca, incluindo diferentes categorias como Estudantes de Graduação, Estudantes de Pós-Graduação, Professores e Funcionários.
- **Empréstimo:** Esta classe gerencia os empréstimos feitos pelos membros, contendo informações sobre quais itens foram emprestados, datas de empréstimo e devolução, entre outros.
- **Renovação e Reserva:** Gerencia as solicitações de renovações e reservas feitas pelos membros para determinados itens.
- **Funcionário:** Representa os funcionários que trabalham na biblioteca, incluindo diferentes níveis de acesso, como Administradores, Atendentes e Gerentes.
- **Relatório:** Esta classe se encarrega de gerar diversos tipos de relatórios sobre a atividade da biblioteca, desde estatísticas de uso até relatórios financeiros de multas.

Com base nessas definições, implemente as relações de Agregação, Composição e Associação entre as classes apresentadas. Para isso, considere:

- **Associação:** É uma relação entre duas classes. Por exemplo, um Membro está associado a vários Empréstimos e um Empréstimo está associado a um único Membro.
- **Agregação:** É uma relação unidirecional entre um todo e suas partes. Por exemplo, a biblioteca é um "todo" e os ItensMultimídia são suas partes. Neste caso, a Biblioteca tem uma agregação com os ItensMultimídia.
- **Composição:** É uma relação forte entre um todo e suas partes, onde a parte não pode existir sem o todo. Por exemplo, um Relatório pode ser composto por várias seções, e essas seções não têm sentido sem o Relatório ao qual pertencem.

Garanta que cada relação esteja corretamente representada em seu código, e que a estrutura permita que a Biblioteca funcione de maneira eficiente e coesa. Lembre-se, você pode apresentar outras classes adicionais desde que estejam dentro do domínio da Biblioteca Universitária Multimídia. A complexidade e o detalhamento dos relacionamentos entre as classes podem variar, mas devem ser justificados dentro do contexto.

1.1 Objetivo

- Seu objetivo é implementar em Java as classes acima mencionadas, modelando corretamente as relações de Agregação, Composição e Associação entre elas. Você deverá identificar onde cada um desses tipos de relacionamento se aplica e implementar adequadamente na sua solução.

1.2 Requisitos

- Utilize construtores, métodos de acesso (*getters* e *setters*) e métodos adicionais conforme necessário.
- Demonstre as relações através de um programa de teste que cria objetos e estabelece as relações entre eles.
- Comente seu código para explicar como você está representando cada relação.

1.3 Dicas

- Para associação, você pode utilizar referências diretas ou coleções (como uma lista) dependendo da cardinalidade da relação.
- Para agregação, você pode utilizar uma coleção para representar a relação "um para muitos", com a distinção de que os objetos agregados podem existir independentemente.
- Para composição, você pode também utilizar uma coleção, mas com a restrição de que os objetos compostos não devem existir fora da entidade composta.

2 Avaliação

- Entrega realizada dentro do prazo estipulado.
- Execução do código.
- Qualidade do código desenvolvido (saída dos dados na tela, tabulação, comentários).
- Desenvolvimento correto dos atributos para cada classe
- Desenvolvimento correto dos tipos de acesso em cada atributo e método.
- Desenvolvimento correto dos *getters* e *setters* nas classes necessárias.
- Seu trabalho será avaliado com base na clareza, coerência e abrangência de suas definições de atributos, tipo de acesso, *getters* e *setters*, e modularidade.

3 Entrega

- Você não precisa criar outro repositório ou outro projeto, a ideia é avançar sobre o repositório que você já tem.
- A entrega do laboratório é realizada exclusivamente via **Github**. Para a submissão no Github, gere um release (tag) com a identificação do laboratório no formato <lab03-RA>. Por exemplo, para o aluno com RA 123456, a tag será: lab03-123456.
- Evite criar *releases* enquanto não tiver certeza que seu código está funcionando como esperado.

- Utilize os horários de laboratório e atendimentos para tirar eventuais dúvidas de submissão e também relacionadas ao desenvolvimento do laboratório.
- Data da entrega: **27/08/2023** até às **23h59**.