

## Exercícios de ProjOO – 03/07/2024

### Instruções:

- Forma de realização: dupla – usar o conceito de *[distributed] pair-programming*;
- Entregar, via Classroom, os JPGs ou GIFs do Diagrama de Classe e os seus respectivos códigos-fonte e de teste (que mostram um exemplo da execução da aplicação);
- Prazo: 14/07/2024. Há também a opção de entrega ao final do recesso, porém com uma redução na pontuação.

## Exercício de Programação: Sistema de Gerenciamento de Biblioteca

### Descrição Geral

Vocês são requisitados a desenvolver um sistema de gerenciamento de biblioteca que permitirá aos usuários (estudantes e funcionários) realizar operações essenciais como **busca de livros, empréstimo de livros, devolução de livros e consulta de informações detalhadas sobre livros e usuários**. O sistema deve ser implementado utilizando Java ou Python e deve incorporar diversas técnicas e padrões de projeto discutidas em sala de aula, como os princípios SOLID.

A escolha da arquitetura, de implementação da interface e do sistema de **persistência de dados fica a critério dos alunos**. Vocês podem, por exemplo, optar por desenvolver uma interface gráfica ou uma interface de linha de comando. Quanto à persistência de dados, podem utilizar bancos de dados reais ou armazenar os dados em memória. Alternativamente, os dados podem ser previamente definidos ou lidos/escritos de/para arquivos em formatos como JSON, TXT, HTML, entre outros.

### Funcionalidades Requeridas

1. **Buscar Livros:** O sistema deve permitir que os usuários realizem buscas por livros utilizando diferentes critérios como **título, autor ou categoria**. O mecanismo de busca deve **adaptar** fontes externas de dados de catalogação.
2. **Emprestar Livros:** Usuários devem ser capazes de emprestar livros que estejam disponíveis na biblioteca. É necessário gerenciar a disponibilidade dos livros e verificar a

elegibilidade dos usuários para realizar empréstimos. As operações principais devem ser fornecidas através de uma interface **simplificada**.

3. **Devolver Livros:** O sistema deve permitir a devolução de livros previamente emprestados, atualizando o status de disponibilidade do livro.

4. **Consultar Informações de Livros:** Usuários devem poder consultar detalhes de livros específicos, incluindo informações como título, autor, categoria e disponibilidade. A estrutura de categorias e subcategorias deve ser gerenciada de forma **hierárquica**.

5. **Consultar Informações de Usuários:** O sistema deve fornecer uma maneira de consultar o histórico de empréstimos de um usuário específico, permitindo ver quais livros foram emprestados e devolvidos. A lógica de diferentes tipos de usuários deve ser **separada**.

## Requisitos de Design

Para implementar as funcionalidades acima, espera-se que vocês apliquem as técnicas de design discutidas no curso, garantindo um código modular, extensível e de fácil manutenção. Abaixo estão algumas orientações sobre a implementação esperada:

1. **Integração com Sistemas Externos:** Crie uma solução que permita a integração de dados de sistemas de catalogação externos de forma transparente para os usuários do sistema.

- **Classe Sugerida:** `ExternalCatalogAdapter`

2. **Interface Simplificada para Operações:** Desenvolva uma interface que forneça acesso simplificado às operações principais do sistema, como busca de livros, empréstimos e devoluções.

- **Classe Sugerida:** `LibraryFacade`

3. **Gerenciamento Hierárquico de Categorias:** Implemente uma estrutura que permita o gerenciamento de livros e suas categorias de forma hierárquica, facilitando a navegação e busca.

- **Classe Sugerida:** `BookCategory`

4. **Abstração e Implementação para Diferentes Usuários:** Separe a lógica de diferentes tipos de usuários, permitindo que operações específicas sejam realizadas de acordo com o tipo de usuário (estudante, professor, funcionário).

- **Classes Sugeridas:** `User`, `StudentUserType`, `TeacherUserType`

5. **Gerenciamento Centralizado de Configurações:** Assegure que as configurações globais do sistema sejam gerenciadas de maneira **única** em toda a aplicação.

- **Classe Sugerida:** `ConfigurationManager`

6. **Notificações aos Usuários:** Implemente um mecanismo de notificação para informar os usuários sobre novas adições ao catálogo ou mudanças no status dos livros.

- **Classe Sugerida:** `BookAvailabilityNotifier`

7. **Coordenação de Interações do Sistema:** Desenvolva uma solução que coordene de forma centralizada as interações entre diferentes componentes do sistema, como busca, empréstimos e devoluções.

- **Classe Sugerida:** `LibraryMediator`

8. **Aprovação de Empréstimos:** Crie um fluxo de aprovação para empréstimos de livros, considerando diversos critérios como disponibilidade do livro, número de livros já emprestados e elegibilidade do usuário. O fluxo de aprovação deve ser gerenciado de forma encadeada.

- **Classes Sugeridas:** `BookAvailabilityHandler`, `UserEligibilityHandler`, `LoanLimitHandler`