

Trabalho Prático 2: Sistema de gerenciamento para bibliotecas

Disciplina: Programação Orientada a Objetos

{*Entregar até o dia 18/05}

Prof. Dr. Lucas C. Ribas

Sumário

1	Descrição	2
1.1	Requisitos	2
1.2	Classes e Interfaces	2
1.3	Exceções	3
1.4	Documentação	3
1.5	Dados	3
1.6	Considerações Adicionais	3
2	Avaliação	3
3	Dúvidas	4

1 Descrição

Este trabalho tem como objetivo desenvolver um sistema de gerenciamento de empréstimos de biblioteca. O sistema deve utilizar os conceitos de herança, polimorfismo, interfaces, classes e métodos abstratos e exceções. Para isso, o sistema deve ser implementado usando a linguagem de programação orientada a objetos Java.

1.1 Requisitos

O sistema de biblioteca deve possuir as seguintes funcionalidades:

1. Cadastro de usuários e itens;
2. Empréstimo e devolução de itens: o sistema gerencia o processo de empréstimo e devolução de itens. Os usuários podem emprestar itens disponíveis, e o sistema registra a data de empréstimo, o prazo de devolução e o usuário responsável. Ao devolver um item, o sistema verifica se o prazo de devolução foi cumprido e, se necessário, aplica multas por atraso.
3. Busca por itens disponíveis: busca pelo nome do item, listando os dados do item como quantidade disponível, etc.
4. Listagem de itens emprestados por usuário: digite a matrícula do usuário para listagem;
5. Aplicação de multas por atraso na devolução.

1.2 Classes e Interfaces

As seguintes classes e interfaces devem ser implementadas no sistema:

- Classe abstrata **Item**: Deve conter os atributos título, autor, ano de publicação, quantidade disponível e quantidade emprestada. Esta classe deve implementar uma interface chamada **Emprestavel**, que define os métodos para empréstimo e devolução.
- Classe **Livro**: Deve herdar da classe **Item** e conter atributos adicionais, como editora e ISBN.
- Classe **Revista**: Deve herdar da classe **Item** e conter atributos adicionais, como volume e número.
- Classe **CD**: Deve herdar da classe **Item** e conter atributos adicionais, como volume e gravadora.
- Classe abstrata **Usuario**: Deve conter os atributos nome, matrícula e lista de itens emprestados. Esta classe deve possuir um método abstrato para calcular a multa por atraso na devolução.
- Classe **Aluno**: Deve herdar da classe **Usuario** e conter atributos adicionais, como curso e período. Deve implementar o método para calcular a multa por atraso na devolução considerando 5,00 reais por dia de atraso.
- Classe **Professor**: Deve herdar da classe **Usuario** e conter atributos adicionais, como departamento e titulação. Deve implementar o método para calcular a multa por atraso na devolução considerando uma taxa de 25% a mais em relação ao Aluno.
- Classe **Emprestimo**: Deve conter os atributos item (referência ao item emprestado), data de empréstimo, data de devolução prevista e data de devolução real. Esta classe pode ser responsável por calcular a multa por atraso na devolução, considerando a data de devolução prevista e a data de devolução real. Quando um usuário empresta um item, um novo objeto da classe **Emprestimo** é criado e adicionado à lista de empréstimos do usuário. Quando o item é devolvido, a data de devolução real é atualizada no objeto **Emprestimo**, e a multa por atraso na devolução, se houver, é calculada.

- Classe **AssesorTecnico**: Deve herdar da classe **Usuario** e conter atributos adicionais, como Seção (Acadêmica, Pos-Graduação, Graduação, Administrativa etc). Deve implementar o método para calcular a multa por atraso na devolução considerando uma taxa de 15% a mais em relação ao Aluno.

1.3 Exceções

O sistema deve tratar, pelo menos, as seguintes exceções:

- Tentativa de empréstimo de um item indisponível;
- Tentativa de devolução de um item não emprestado;
- Tentativa de cadastro de um usuário ou item com informações inválidas.

1.4 Documentação

Documentar o projeto, explicando o propósito e o funcionamento das classes, interfaces, métodos e propriedades criadas. A documentação pode ser incluída como comentários no código-fonte e/ou em um documento separado, como um relatório ou manual do usuário.

1.5 Dados

Opcionalmente, você pode usar os dados do arquivo `items.txt` em anexo. Cada linha desse arquivo é um tipo de item (CD, Livro, Revista) com diferentes campos:

- Livro; Título: Título do Livro; Autor: Autor do Livro; Ano de Publicação; Quantidade Disponível; Editora do Livro; ISBN
- Revista; Título da Revista; Autor da Revista; Ano de Publicação; Quantidade Disponível; Volume; Número
- CD; Título do CD; Compositor do CD; Ano de Publicação; Quantidade Disponível; Volume Gravadora do CD

O uso é opcional, você pode modificar ou colocar campos diferentes também. Para leitura adapte o código do primeiro trabalho.

1.6 Considerações Adicionais

Para aprimorar o projeto, vocês podem considerar incluir funcionalidades adicionais, como:

- Permitir que o usuário crie uma conta e faça login no sistema para gerenciar suas reservas
- Implementar uma interface gráfica de usuário (GUI) para melhorar a experiência do usuário, usando bibliotecas como JavaFX ou Swing

2 Avaliação

- Este trabalho é **individual**. Isso significa que qualquer tipo de plágio será penalizado com nota **ZERO**.
- Posteriormente à entrega do trabalho, poderei solicitar que alguns de vocês expliquem o código. Essa entrevista também poderá influenciar na nota final.

- Seu pagamento será em moedas do tipo Nota :). A nota atribuída ao trabalho será de 0 a 10 e fará parte da média de trabalhos realizados ao longo do semestre. O peso dessa média foi definido no início da disciplina.
- Implementações extras, como a classe Extrato, e a utilização de boas práticas de programação, como documentação JavaDoc, contarão como ponto extra.
- A entrega deve ser feita via classroom. Você deverá anexar todo o projeto em formato .zip.
- Deadline: **18/05/2023**. Não terá adiamentos e não serão aceitos trabalhos após o prazo de entrega.

3 Dúvidas

Qualquer dúvida ou dificuldade, não hesite em me procurar. O mais importante é que você aprenda.

Meu e-mail é lucas.ribas@unesp.br e meu gabinete é o 12 no DCCE (envie um e-mail antes para checar a disponibilidade).