

Melhorias no Gerenciador de Biblioteca – Tipos Abstratos de Dados

(APP) Atividade Prática com Prazo de Entrega e Com Consulta – Peso 1

Objetivos Pedagógicos da Atividade

1. Identificar o conceito de interface e sua finalidade na linguagem **Java**;
2. Entender como a interface promove o polimorfismo e melhora a flexibilidade do sistema;
3. Implementar a interface **ILivro** e aplicá-la a diferentes tipos de livros usando reuso através de herança e polimorfismo;
4. Analisar as diferenças entre o uso de uma classe concreta e uma interface na manipulação de livros;
5. Compreender se a nova arquitetura com interface melhora a extensibilidade e coesão do código;
6. Criar uma arquitetura extensível com múltiplos tipos de livro e integração com a biblioteca existente.

Melhorias Para Implementação

1. Criar um Tipo Abstrato de Dado (TAD) denominado **ILivro**. O TAD deve ser uma especificação do que pode ser um livro na biblioteca, portanto, deve conter as assinaturas dos métodos públicos da classe **Livro**, exceto o construtor, assim será definido o contrato que define semanticamente o que é um livro genérico nesta biblioteca - valor (1,0) ponto.
2. Alterar a assinatura do método **public void adicionarLivro(Livro livro)** na interface **IBiblioteca** para que ela passe a usar a interface **ILivro** como tipo do parâmetro **livro** - valor (0,5) ponto.
3. Alterar a assinatura do método **public Livro buscarLivro(String titulo)** na interface **IBiblioteca** para que ela passe a usar a interface **ILivro** como tipo do retorno do método - valor (0,5) ponto.

4. Refatorar a classe **Biblioteca** para que ela passe a funcionar usando a interface **ILivro** - valor (2,0) pontos.
- a. Não pode adicionar novos métodos à classe **Biblioteca**;
 - b. Não pode alterar o nome dos métodos já existentes na classe **Biblioteca**.
5. Você foi contratado para adicionar novas funcionalidades ao projeto de gerenciador de bibliotecas. O contratante solicitou que no gerenciador ele possa também adicionar, remover, emprestar, devolver e listar livros impressos, livros eletrônicos e livros em áudio - valor (6,0) pontos.
- a. Os livros impressos, eletrônicos e em áudio devem ter suas próprias classes (**LivroImpresso**, **LivroEletronico** e **LivroEmAudio**) e devem implementar a interface **ILivro** - valor (0,5) ponto.
 - b. Os livros impressos, eletrônicos e em áudio possuem em comum os atributos **titulo**, **autor**, **anoPublicacao** e **disponivel** que já estão implementados na classe **Livro** e devem ser reusados para evitar duplicação de código - valor (0,5) ponto.
 - c. A classe **LivroImpresso** - valor (1,5) pontos:
 - i. Deve ter o atributo **paginas** do tipo **int** e seus respectivos métodos **get** e **set**;
 - ii. O construtor da classe **LivroImpresso** deve receber o atributo **paginas** como parâmetro e atribuir ao atributo **paginas**.
 - iii. Deve sobrescrever o método **public String toString()** para imprimir no seguinte exemplo de formato:

- **Livro Impresso, de Pedro (2000) Páginas: 200 - Disponível**

Obs.: Livro Impresso é o de título do livro, Pedro é o autor do Livro, 2000 é o ano de publicação do livro, 200 é o número de páginas do livro e Disponível é a indicação se o livro está disponível ou não para empréstimo.

d. A classe **LivroEletronico** - valor (1,5) pontos:

- i. Deve ter o atributo **tamanho** do tipo **int** e seus respectivos métodos **get** e **set**;
- ii. O construtor da classe **LivroEletronico** deve receber o atributo **tamanho** como parâmetro e atribuir ao atributo **tamanho**.
- iii. Deve sobrescrever o método **public String toString()** para imprimir no seguinte exemplo de formato:

- **Livro Eletrônico, de Maria (2004) Tamanho: 39 - Disponível**

Obs.: Livro Eletrônico é o de título do livro, Maria é a autora do Livro, 2004 é o ano de publicação do livro, 39 é o tamanho em megabytes do livro e Disponível é a indicação se o livro está disponível ou não para empréstimo.

e. A classe **LivroEmAudio** – valor (1,5) pontos:

- i. Deve ter o atributo **tempo** do tipo **int** e seus respectivos métodos **get** e **set**;
- ii. O construtor da classe **LivroEmAudio** deve receber o atributo **tempo** como parâmetro e atribuir ao atributo **tempo**.
- iii. Deve sobrescrever o método **public String toString()** para imprimir no seguinte exemplo de formato:

- **Livro em Áudio, de Jose (2025) Tempo: 2 - Disponível**

Obs.: Livro em Áudio é o de título do livro, Jose é o autor do Livro, 2025 é o ano de publicação do livro, 2 é o tempo em horas do livro e Disponível é a indicação se o livro está disponível ou não para empréstimo.

f. A classe **App** deve ser alterada para disponibilizar as opções de adicionar livros impressos, eletrônicos e em áudio - valor (0,5) ponto.

Exemplo de Resultado Esperado – Ao Listar Acervo Após Adição de Alguns Livros

Após a adição de três livros no acervo da biblioteca (um impresso, um eletrônico e um em áudio) o resultado da ação de listar acervo deve ser algo semelhante ao exemplo abaixo.

```
MENU BIBLIOTECA
1. Adicionar livro impresso
2. Adicionar livro eletrônico
3. Adicionar livro em audio
4. Remover livro
5. Emprestar livro
6. Devolver livro
7. Listar acervo
0. Sair
Escolha uma opção: 7
Livros disponíveis no acervo:
- Livro Impresso, de Pedro (2000) Páginas: 200 - Disponível
- Livro Eletronico, de Maria (2004) Tamanho: 39 - Disponível
- livro em Audio, de Jose (2025) Tempo: 2 - Disponível
```