

Exercício 1: Sistema de Gerenciamento de Biblioteca

Descrição do Estudo de Caso

Uma biblioteca pública deseja informatizar o gerenciamento de seus livros e empréstimos. O sistema deve permitir ao bibliotecário:

- **Cadastrar** novos livros no acervo.
- **Registrar** empréstimos de livros para os usuários.
- **Consultar** a disponibilidade de um livro específico.

Você foi contratado para desenvolver uma aplicação em TypeScript que atenda a essas necessidades.

Tarefas

1. **Crie uma classe `Livro`** que possua:
 - Propriedades:
 - `codigo` (number): identificador único do livro.
 - `titulo` (string): título do livro.
 - `autor` (string): autor do livro.
 - `disponivel` (boolean): indica se o livro está disponível para empréstimo.
 - Construtor que inicializa todas as propriedades acima.
2. **Crie uma classe `Biblioteca`** que gerencie os livros:
 - Propriedade privada:
 - `acervo` (array de `Livro`): lista de livros disponíveis na biblioteca.
 - Métodos:
 - `adicionarLivro(livro: Livro): void` - Adiciona um novo livro ao acervo.
 - `registrarEmprestimo(codigo: number): void` - Marca o livro com o código especificado como indisponível.
 - `consultarDisponibilidade(codigo: number): boolean` - Retorna `true` se o livro estiver disponível, ou `false` caso contrário.
3. **Implemente funções** para testar o sistema:
 - **Função para cadastrar livros:** Crie pelo menos três instâncias de `Livro` e adicione-as ao acervo usando o método `adicionarLivro`.
 - **Função para registrar empréstimo:** Realize o empréstimo de um dos livros cadastrados.

- **Função para consultar disponibilidade:** Verifique a disponibilidade de um livro pelo código e imprima o resultado no console.

Requisitos Técnicos

- Utilize os conceitos de **classe** e **função** em TypeScript.
- Use modificadores de acesso apropriados (**public**, **private**) para encapsulamento.
- Tipifique todas as variáveis e retornos de funções explicitamente.
- Trate possíveis erros, como tentativas de emprestar ou consultar livros inexistentes.

Objetivos de Aprendizado

- Praticar a definição e utilização de **classes** em TypeScript.
- Implementar e invocar **funções** para manipular objetos e classes.
- Aplicar conceitos de encapsulamento e tipagem estática.
- Simular um cenário real de gerenciamento de biblioteca em uma aplicação.

Entrega

- Código-fonte em TypeScript com a implementação das classes e funções solicitadas.
- Comentários no código explicando o funcionamento de cada parte.
- Um arquivo README explicando como executar o programa e os testes realizados.