**OPÇÃO 01 :**

Descrição do Projeto: **Sistema de Gerenciamento de Biblioteca**

Objetivo:

Desenvolver um sistema de gerenciamento de biblioteca em Java que permita aos usuários realizar operações básicas, como cadastrar, buscar, atualizar e excluir livros, além de realizar empréstimos e devoluções.

Funcionalidades Principais:

1. Cadastro de Livros:
   * Permitir o cadastro de novos livros no sistema, incluindo informações como título, autor, editora, ano de publicação, ISBN, e quantidade disponível.
2. Busca de Livros:
   * Permitir a busca de livros por título, autor, ou ISBN, fornecendo informações detalhadas sobre os livros encontrados.
3. Atualização de Livros:
   * Permitir a atualização das informações de livros cadastrados, como título, autor, editora, etc.
4. Exclusão de Livros:
   * Permitir a exclusão de livros do sistema.
5. Empréstimo de Livros:
   * Permitir que usuários façam empréstimos de livros disponíveis na biblioteca, registrando informações como usuário, data de empréstimo e data de devolução prevista.
   * Verificar a disponibilidade do livro antes de realizar o empréstimo.
6. Devolução de Livros:
   * Permitir que usuários realizem a devolução de livros emprestados, registrando a data de devolução efetiva.
   * Atualizar a quantidade disponível do livro após a devolução.

Arquitetura do Projeto:

O projeto será implementado em Java utilizando o paradigma de Programação Orientada a Objetos (POO). Serão definidas as seguintes classes principais:

* Livro:

A classe Livro representará as informações de cada livro, incluindo atributos como título, autor, editora, ano de publicação, ISBN e quantidade disponível. Esta classe também conterá métodos para acessar e modificar esses atributos.

| class Livro {  private String titulo;  private String autor;  private String editora;  private int anoPublicacao;  private String ISBN;  private int quantidadeDisponivel;  // Construtor  public Livro(String titulo, String autor, String editora, int anoPublicacao, String ISBN, int quantidadeDisponivel) {  this.titulo = titulo;  this.autor = autor;  this.editora = editora;  this.anoPublicacao = anoPublicacao;  this.ISBN = ISBN;  this.quantidadeDisponivel = quantidadeDisponivel;  }  // Métodos Getters e Setters  // ...  // Métodos para empréstimo e devolução  // ...  } |
| --- |

Observação: Nesse momento, os métodos para empréstimo e devolução podem ser definidos como futuras implementações, considerando a lógica de negócio específica da biblioteca.

* Biblioteca:

A classe Biblioteca será responsável por gerenciar os livros cadastrados e realizar operações como adicionar, buscar, atualizar e excluir livros.

| import java.util.ArrayList;  import java.util.List;  class Biblioteca {  private List<Livro> acervo;  // Construtor  public Biblioteca() {  this.acervo = new ArrayList<>();  }  // Métodos para operações de livros (adicionar, buscar, atualizar, excluir)  // ...  // Métodos para empréstimo e devolução  // ...  } |
| --- |

Classe Teste:

A classe de teste será utilizada para verificar o funcionamento correto do sistema, instanciando objetos da classe Biblioteca e realizando operações de teste, como cadastro de livros, busca, empréstimo, devolução, etc.

| public class TesteBiblioteca {  public static void main(String[] args) {  // Criando uma nova biblioteca  Biblioteca biblioteca = new Biblioteca();  // Adicionando alguns livros ao acervo  Livro livro1 = new Livro("Java: Como Programar", "Deitel & Deitel", "Pearson", 2020, "978-8576059029", 5);  Livro livro2 = new Livro("Clean Code", "Robert C. Martin", "Prentice Hall", 2008, "978-0132350884", 3);  biblioteca.adicionarLivro(livro1);  biblioteca.adicionarLivro(livro2);  // Realizando operações de teste (busca, empréstimo, devolução, etc.)  // ...  // Exibindo o acervo da biblioteca após as operações de teste  System.out.println("Acervo da Biblioteca:");  biblioteca.exibirAcervo();  }  } |
| --- |

Observação: Nesse exemplo, o código para adicionar livros, realizar operações de busca, empréstimo, devolução, etc., na classe Biblioteca e seus métodos associados não estão implementados. Essas são partes do projeto que devem ser desenvolvidas de acordo com os requisitos específicos do sistema.

**OPÇÃO 2 :**

Você deve criar **OS 3 PROJETOS ABAIXO**  ( Herança) com as respectivas classe Pai ABSTRATA e as respectivas classes MAIN/TESTE

**Projeto 1- Moradia**

Moradia - classe pai ABSTRATA

casa e apartamento - classes filhas

Classe Principal - classe que contém o método main

**Projeto2 - Game**

Personagem - classe pai ABSTRATA

Arqueiros, Cavaleiro, Mago - classes filhas

Classe Principal - classe que contém o método main

**Projeto 3 - Você deve idealizar - invente 🙂**

você que decide quem serão as classes

OBS: Para todos os projetos, adicionar a impressão utilizando o método toString().

A captura dos dados será pelo teclado - Scanner();

Cadastre vários objetos utilizando um List/ArrayList