



Cenário: Sistema de Biblioteca para Empréstimo de Livros

Objetivo: Trabalhar os conceitos de Herança e Polimorfismo em um cenário de gerenciamento de uma biblioteca. Nome do projeto: **PrjBiblioteca**.

Enunciado:

A biblioteca de uma faculdade deseja um sistema para controlar o empréstimo de livros para alunos e professores. Para isso, o sistema deve ser capaz de registrar os detalhes dos empréstimos e devoluções de exemplares dos livros.

Existem três tipos de usuários da biblioteca: **Alunos**, **Professores** e **Funcionários**. Cada um desses usuários possui características específicas e, portanto, classes diferentes que herdam uma classe base genérica chamada **Usuario**.

As classes devem ser modeladas e implementadas conforme os exercícios a seguir:

1. **Crie um novo projeto Java chamado PrjBiblioteca.** A classe que contém o método **main** deve ser chamada de **GerenciarBiblioteca**.
2. **Crie uma classe Java chamada Usuario** com os seguintes atributos privados:
 - idUsuario (int)
 - nome (String)
 - email (String)

Defina os métodos getters e setters para prover acesso a esses campos. Defina o método construtor para receber os parâmetros necessários.

3. **Crie uma subclasse de Usuario chamada Aluno**, com os seguintes atributos:
 - matricula (String)
 - curso (String)

Defina os campos como privados e crie os métodos getters e setters para prover acesso. Defina o método construtor para receber os parâmetros necessários.

4. **Crie uma subclasse de Usuario chamada Professor**, com os seguintes atributos:
 - departamento (String)
 - titulação (String)

Defina os campos como privados e crie os métodos getters e setters para prover acesso. Defina o método construtor para receber os parâmetros necessários.

5. **Crie uma subclasse de Usuario chamada Funcionario**, com os seguintes atributos:
 - cargo (String)
 - setor (String)

Defina os campos como privados e crie os métodos getters e setters para prover acesso. Defina o método construtor para receber os parâmetros necessários.

6. **Crie uma nova classe chamada Livro**, com os seguintes atributos privados:



- idLivro (int)
- titulo (String)
- autor (String)
- disponivel (boolean)

Defina os métodos getters e setters conforme o padrão. Defina o método construtor para receber os parâmetros necessários.

7. Crie uma classe chamada RegistroEmprestimo, com os seguintes atributos privados:

- idEmprestimo (int)
- usuario (tipo Usuario)
- livro (tipo Livro)
- dataEmprestimo (LocalDate)
- dataDevolucao (LocalDate)

Defina os métodos getters e setters para prover acesso aos campos. Defina o método construtor para receber os parâmetros necessários.

8. Crie uma classe chamada ControleEmprestimo que deve ter dois métodos principais:

- registraEmprestimo: que recebe um objeto do tipo Usuario e um objeto do tipo Livro. Ele deve registrar o empréstimo do livro e retornar uma mensagem contendo os detalhes do empréstimo, incluindo a data. Veja um exemplo de saída no final deste material.
- registraDevolucao: que recebe um objeto do tipo Usuario e um objeto do tipo Livro. Ele deve registrar a devolução do livro e retornar uma mensagem contendo os detalhes da devolução, incluindo a data. Veja um exemplo de saída no final deste material.

9. Classe GerenciarBiblioteca:

- No método main, instancie um Aluno, um Professor e um Funcionario e popule os dados.
- Crie também alguns objetos Livro e popule os dados.
- Registre o empréstimo dos livros para cada um dos usuários e mostre no console os detalhes dos empréstimos.
- Registre a devolução dos livros para cada um dos usuários e mostre no console os detalhes das devoluções.

Diagrama de Classes

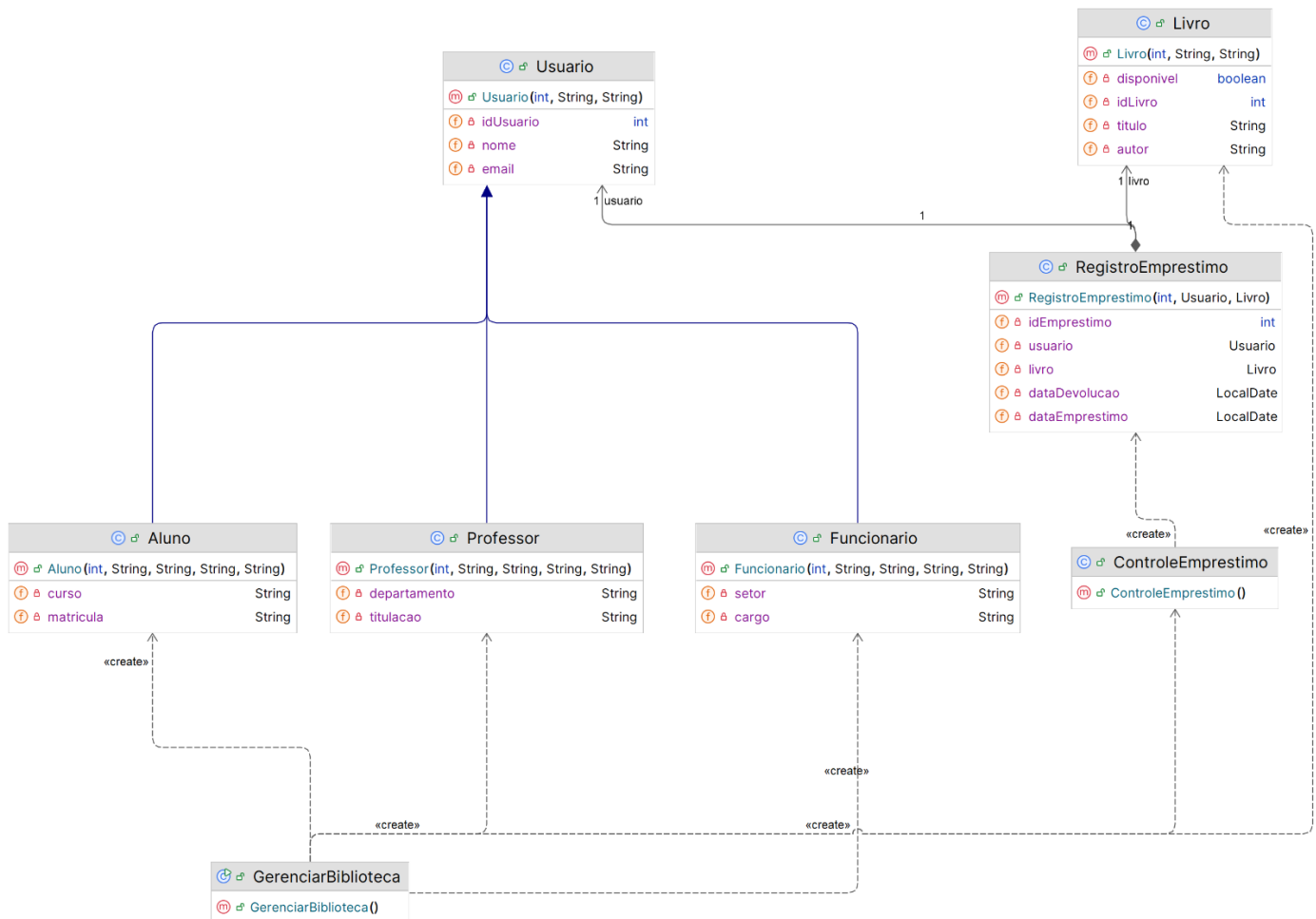
Vou descrever um diagrama de classes para representar as relações de herança e associações:

- **Usuario (Classe Base)**
 - Atributos: idUsuario, nome, email
 - Métodos: getters/setters
 - **Aluno (Subclasse de Usuario)**
 - Atributos: matricula, curso
 - Métodos: getters/setters
 - **Professor (Subclasse de Usuario)**
 - Atributos: departamento, titulação
 - Métodos: getters/setters
 - **Funcionario (Subclasse de Usuario)**
 - Atributos: cargo, setor
 - Métodos: getters/setters
- **Livro**
 - Atributos: idLivro, titulo, autor, disponivel
 - Métodos: getters/setters
- **RegistroEmprestimo**
 - Atributos: idEmprestimo, usuario, livro, dataEmprestimo, dataDevolucao

- Métodos: getters/setters
- **ControleEmprestimo**
 - Métodos: registraEmprestimo(usuario: Usuario, livro: Livro), registraDevolucao(usuario: Usuario, livro: Livro)

Diagrama de Classes em Imagem:

Abaixo um diagrama visual para facilitar a compreensão dos relacionamentos.



Explicação do Diagrama:

- **Herança:** A classe Usuario é a base para Aluno, Professor e Funcionario, representando o conceito de reutilização de atributos e métodos comuns.
- **Associações:**
 - A classe RegistroEmprestimo está associada às classes Usuario e Livro, indicando que um registro de empréstimo envolve um usuário específico e um livro específico.
 - A classe ControleEmprestimo possui métodos que utilizam instâncias da classe Usuario e Livro para registrar as operações de empréstimo e devolução.

Este exercício permitirá que os alunos pratiquem conceitos de **herança, polimorfismo, associação e composição**.

Exemplo de Saída – Execução:

Empréstimo registrado para: Maria Silva

Livro: Java para Iniciantes

Data do Empréstimo: 2024-10-01

Empréstimo registrado para: Dr. João Oliveira

Livro: Estruturas de Dados

Data do Empréstimo: 2024-10-01

Devolução registrada para: Maria Silva

Livro: Java para Iniciantes

Data da Devolução: 2024-10-01

Devolução registrada para: Dr. João Oliveira

Livro: Estruturas de Dados

Data da Devolução: 2024-10-01