RECUPERAÇÃO

Programação Orientada a Objetos

Felipe da Silva Andrade

Instituto Federal do Paraná - Campus Pinhais
Bacharelado em Ciência da Computação (BCC)

Janeiro de 2025

Sumário

1 Introdução

2 Especificações do Sistema

2.1 Requisitos

- Linguagem de programação utilizada: Java.
  - Banco de dados:
  - Aplicação dos conceitos de encapsulamento, herança, polimorfismo e abstração.
  - Utilização de pelo menos dois padrões de projetos.

2.2 Regras

Um usuário pode realizar até 3 empréstimos simultâneos (aluno) ou até 5 (professor). • A devolução de um livro deve atualizar a disponibilidade do mesmo. • O sistema deve permitir cadastrar, listar, editar e remover os livros, usuários e empréstimos realizados.

3 Aplicação dos Conceitos de Orientação a Objetos

3.1 Encapsulamento

O encapsulamento no projeto se resume ao uso de protected no para delimitar o acesso das variaveis pelos métodos de outras classes.

Listing 1: Exemplo de Encapsulamento

```
1
2  public abstract class UsuarioModel{
3
4      protected String CPF;
5      protected String senha;
6      protected String nome;
7      protected String sobreNome;
8      ....
9
10
11 }
```

3.2 Herança

A herança foi planejada de forma que todos os atributos em comum do aluno e do professor ficassem numa classe nomeada de usuário, dessa forma implementando herança e a classe abstrata.

Listing 2: Exemplo de Herança

```
1
2
3  public class AlunoModel extends UsuarioModel{
4
5      protected String matricula;
```

```
6
7       public AlunoModel(){} // Construtor
8
9       public AlunoModel(String matricula, String CPF, String senha, String
            nome, String sobreNome){
10      this.matricula = matricula;
11      ....
12
13
14
15  }
```

3.3 Polimorfismo

O uso de polimorfismo no projeto é quase que exclusivo na implementação da inferface nas classes Controller visto que várias das classes Controller usam @Override nas funções criar, remover, listar/atualizar e deletar.

Listing 3: Exemplo de Polimorfismo

```
1
2
3   public class LivroController implements ICrud {
4       ...
5       @Override
6       public void criar() {
7           view.cadastrarLivro(model);
8           banco.cadastroLivros(model);
9       }
10      ...
11
12  }
13  }
```

4 Padrões de Projeto Utilizados

A principio, o padrão de projeto mais notável utilizado no projeto é o MVC/DAO, possivelmente tendo algumas decisões que usam ideia do modelo padrão

5 Banco de Dados

O Banco de dados é formado de 4 tabelas, uma para aluno e uma para professor, tendo ambas CPF (chave primaria), nome, sobrenome e senha como atributos em comum e matricula (aluno) e siape (professor) como atributos únicos. A terceira tabela comporta os dados dos livros sendo apenas o ID, uma variavel do tipo inteiro auto incremental (chave primária) e o nome, a quarta e última tabela faz a ligação entre as 3 primeiras, tendo 3 colunas, uma registra o cpf do aluno no caso do mesmo realizar o empréstimo, a segunda segue a mesma lógica mas com o cpf do professor e a terceira e última marca o ID do livro emprestado.

6 Dificuldades

A implementação da funções de edição e listar nas classes BancoAluno e BancoProfessor foram bem problematicas, especialmente a de listar que quando formatava os dados no terminal, grande parte aparecia "null", coisa que não foi possível ajustar, a abstração e implementação da lógica de empréstimo foi muito desafiadora de forma que não foi possível incluir a função de limite de empréstimo por usuário.

7 Conclusão

No final, o projeto foi parcialmento concluido, pois não foi possível implementar na parte de empréstimo, as delimitações requisitadas por conta do tempo e inexperiência, a listagem dos dados do professor e aluno, no mais, o projeto incorporou quase todos os requisitos.

Referências

Conector do SQL pro Java

Site do Professor William

Modelos de projeto