```
/** * Esta classe representa a interface grA¡fica para alterar informaA§Aµes de um livro. * Os dados sA£o obtidos de uma tabela do banco de dados. * O cÃ³digo-fonte foi descompilado de um arquivo .class usando o decompilador FernFlower. * * @author [Seu Nome] * @version 1.0 * @since 2024-06-16 *///
Source code is decompiled from a .class file using FernFlower decompiler.
package view;
import java.awt.Component;
import java.awt.LayoutManager;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JTextField;
import model.ConnectionFactory;
public class AlterarLivroView extends JFrame { private final JTextField idField; private final JTextField tituloField; private final JTextField autorField; private final JTextField generoField; private final JTextField quantidadeField; private final JButton buscarButton; private final JButton atualizarButton; private final ListarLivrosView listarLivrosView; /** * Construtor da classe AlterarLivroView. * * @param listarLivrosView A instÃ¢ncia da ListarLivrosView associada. */ public AlterarLivroView(ListarLivrosView listarLivrosView) { this.listarLivrosView = listarLivrosView; this.setTitle("Alterar Livro"); this.setSize(400, 350); this.setDefaultCloseOperation(2); this.setLocationRelativeTo((Component)null); this.setLayout((LayoutManager)null); JLabel idLabel = new JLabel("ID:"); idLabel.setBounds(20, 20, 100, 25); this.add(idLabel); this.idField = new JTextField(); this.idField.setBounds(120, 20, 200, 25); this.add(this.idField); this.buscarButton = new JButton("Buscar"); this.buscarButton.setBounds(120, 60, 100, 25); this.add(this.buscarButton); JLabel tituloLabel = new JLabel("T\u00edtulo:"); tituloLabel.setBounds(20, 100, 100, 25); this.add(tituloLabel); this.tituloField = new JTextField(); this.tituloField.setBounds(120, 100, 200, 25); this.add(this.tituloField); JLabel autorLabel = new JLabel("Autor:"); autorLabel.setBounds(20, 140, 100, 25); this.add(autorLabel); this.autorField = new JTextField(); this.autorField.setBounds(120, 140, 200, 25); this.add(this.autorField); JLabel generoLabel = new JLabel("G\u00eanero:"); generoLabel.setBounds(20, 180, 100, 25); this.add(generoLabel); this.generoField = new JTextField(); this.generoField.setBounds(120, 180, 200, 25); this.add(this.generoField); JLabel quantidadeLabel = new JLabel("Quantidade:"); quantidadeLabel.setBounds(20, 220, 100, 25); this.add(quantidadeLabel); this.quantidadeField = new JTextField(); this.quantidadeField.setBounds(120, 220, 200, 25); this.add(this.quantidadeField); this.atualizarButton = new JButton("Atualizar"); this.atualizarButton.setBounds(120, 260, 100, 25); this.add(this.atualizarButton); this.buscarButton.addActionListener(new 1(this)); this.atualizarButton.addActionListener(new 2(this)); } /** * Busca as informaÃ§Ãµes do livro a ser alterado. */ private void buscarLivro() { int id = Integer.parseInt(this.idField.getText()); try { Throwable var2 = null; Object var3 = null; try { Connection conn = ConnectionFactory.getConnection(); try { String sql = "SELECT * FROM livro WHERE id = ?"; PreparedStatement stmt = conn.prepareStatement(sql); stmt.setInt(1, id); ResultSet rs = stmt.executeQuery(); if (rs.next()) { this.tituloField.setText(rs.getString("titulo")); this.autorField.setText(rs.getString("autor")); this.generoField.setText(rs.getString("genero")); this.quantidadeField.setText(rs.getString("quantidade")); } else { JOptionPane.showMessageDialog(this, "Livro n\u00e3o encontrado."); } } finally { if (conn != null) { conn.close(); } } } catch (Throwable var15) { if (var2 == null) { var2 = var15; } else if (var2 != var15) { var2.addSuppressed(var15); } throw var2; } } catch (Exception var16) { var16.printStackTrace(); JOptionPane.showMessageDialog(this, "Erro ao conectar ao banco de dados.", "Erro", 0); } } /** * Atualiza as informaÃ§Ãµes do livro no banco de dados. */ private void atualizarLivro() { int id = Integer.parseInt(this.idField.getText()); String titulo = this.tituloField.getText(); String autor = this.autorField.getText(); String genero = this.generoField.getText(); int quantidade = Integer.parseInt(this.quantidadeField.getText()); try { Throwable var6 = null; Object var7 = null; try { Connection conn = ConnectionFactory.getConnection(); try { String sql = "UPDATE livro SET titulo = ?, autor = ?, genero = ?, quantidade = ? WHERE id = ?"; PreparedStatement stmt = conn.prepareStatement(sql); stmt.setString(1, titulo); stmt.setString(2, autor); stmt.setString(3, genero); stmt.setInt(4, quantidade); stmt.setInt(5, id); stmt.executeUpdate(); JOptionPane.showMessageDialog(this, "Livro atualizado com sucesso."); this.listarLivrosView.carregarLivros(); } finally { if (conn != null) { conn.close(); } } } catch (Throwable var18) { if (var6 == null) { var6 = var18; } else if (var6 != var18) { var6.addSuppressed(var18); } throw var6; } } catch (Exception var19) { var19.printStackTrace(); JOptionPane.showMessageDialog(this, "Erro ao conectar ao banco de dados.", "Erro", 0); } } }
```