

Todos os métodos identificados no script TelaPrincipal.java:

Métodos da Classe TelaPrincipal

1. Construtores

- `public TelaPrincipal()` - Construtor da classe que inicializa os componentes e configura a tabela.

2. Métodos de Ação dos Botões

- `private void btnInserirActionPerformed(java.awt.event.ActionEvent evt)` - Lida com a inserção de novos registros.
- `private void btnAtualizarActionPerformed(java.awt.event.ActionEvent evt)` - Atualiza um registro existente.
- `private void btnExcluirActionPerformed(java.awt.event.ActionEvent evt)` - Exclui um registro selecionado.
- `private void btnRelatorioActionPerformed(java.awt.event.ActionEvent evt)` - Gera um relatório com estatísticas.

3. Métodos de Atualização da Interface

- `private void atualizarTabela()` - Atualiza os dados exibidos na tabela.
- `private void carregarDadosSelecionados()` - Carrega os dados da linha selecionada nos campos de texto.
- `private void limparCampos()` - Limpa os campos de entrada.

4. Métodos de Cálculo Estatístico

- `private double calcularMedia(double[] valores)` - Calcula a média de um array de double.
- `private int calcularMedia(int[] valores)` - Calcula a média de um array de int.
- `private double encontrarMaior(double[] valores)` - Encontra o maior valor em um array de double.
- `private double calcularSoma(double[] valores)` - Calcula a soma de um array de double.

5. Métodos Auxiliares

- `private void exibirRelatorio(Object[][] dados)` - Exibe o relatório em uma nova janela.
- `private void initComponents()` - Método gerado pelo NetBeans para inicializar os componentes Swing (não deve ser modificado manualmente).

6. Método main (Ponto de Entrada)

- `public static void main(String args[])` - Inicia a aplicação.

Métodos da Superclasse (javax.swing.JFrame)

Além dos métodos acima, a classe herda métodos de JFrame, como:

- `setDefaultCloseOperation()`
- `setTitle()`
- `setVisible()`
- `pack()`

Resumo

- **Métodos de CRUD:** `btnInserirActionPerformed`, `btnAtualizarActionPerformed`, `btnExcluirActionPerformed`.
- **Métodos de Relatório:** `btnRelatorioActionPerformed`, `exibirRelatorio`.
- **Métodos de Cálculo:** `calcularMedia`, `encontrarMaior`, `calcularSoma`.
- **Métodos de Interface:** `atualizarTabela`, `carregarDadosSelecionados`, `limparCampos`.

Esses métodos trabalham em conjunto para fornecer uma interface funcional de cadastro de pessoas com operações básicas de banco de dados e relatórios.

Aqui estão os métodos transformados em **Portugol** (alto nível de compreensão):

1. atualizarTabela()

```
função atualizarTabela()
início
    // Limpa a tabela atual
    modeloTabela.limparDados()

    // Obtém a lista de pessoas do banco de dados
    listaPessoas <- PessoaDAO.listar()

    // Para cada pessoa na lista, adiciona uma linha na tabela
    para cada pessoa em listaPessoas faça
        modeloTabela.adicionarLinha([
            pessoa.getId(),
            pessoa.getNome(),
            pessoa.getIdade(),
            pessoa.getSalario()
        ])
    fim_para
fim
```

2. carregarDadosSelecionados()

```
função carregarDadosSelecionados()
início
    // Verifica se uma linha foi selecionada
    linhaSelecionada <- tabelaPessoas.getLinhaSelecionada()

    se linhaSelecionada >= 0 então
        // Obtém os dados da linha selecionada
        id <- tabelaPessoas.getValor(linhaSelecionada, 0)
        nome <- tabelaPessoas.getValor(linhaSelecionada, 1)
        idade <- tabelaPessoas.getValor(linhaSelecionada, 2)
        salario <- tabelaPessoas.getValor(linhaSelecionada, 3)

        // Preenche os campos na interface
        campoNome.setTexto(nome)
        campoidade.setTexto(idade)
        campoSalario.setTexto(salario)
    fim_se
fim
```

3. limparCampos()

```
função limparCampos()
início
    campoNome.setTexto("")
    campoldade.setTexto("")
    campoSalario.setTexto("")
fim
```

4. calcularMedia(double[] valores)

```
função calcularMedia(valores: real[]): real
início
    soma <- 0.0

    para cada valor em valores faça
        soma <- soma + valor
    fim_para

    retorne soma / tamanho(valores)
fim
```

5. calcularMedia(int[] valores)

```
função calcularMedia(valores: inteiro[]): inteiro
início
    soma <- 0

    para cada valor em valores faça
        soma <- soma + valor
    fim_para

    retorne soma / tamanho(valores)
fim
```

6. encontrarMaior(double[] valores)

```
função encontrarMaior(valores: real[]): real
início
    maior <- valores[0]

    para cada valor em valores faça
        se valor > maior então
            maior <- valor
        fim_se
    fim_para

    retorne maior
fim
```

7. calcularSoma(double[] valores)

```
função calcularSoma(valores: real[]): real
início
    soma <- 0.0

    para cada valor em valores faça
        soma <- soma + valor
    fim_para

    retorne soma
fim
```

8. exibirRelatorio(Object[][] dados)

```
função exibirRelatorio(dados: matriz[][])
início
    // Cria uma nova janela para o relatório
    janelaRelatorio <- nova Janela("Relatório Completo")

    // Cria uma tabela para exibir os dados
    tabelaRelatorio <- nova Tabela(dados, ["", "", "", ""])

    // Impede a edição dos dados
    tabelaRelatorio.bloquearEdicao()

    // Adiciona a tabela em um painel de rolagem
    painelRolagem <- novo PainelRolagem(tabelaRelatorio)

    // Configura a janela
    janelaRelatorio.adicionar(painelRolagem)
    janelaRelatorio.definirTamanho(600, 400)
    janelaRelatorio.centralizar()
    janelaRelatorio.mostrar()
fim
```

Observações:

1. **Portugol** é uma linguagem de pseudocódigo em português, então:
 - <- representa atribuição.
 - se ... então ... fim_se equivale a if ... { ... }.
 - para cada ... faça ... fim_para equivale a um for ou foreach.
 - função equivale a método em Java.
2. **Adaptações:**

- Substituí `JTable`, `DefaultTableModel`, etc., por termos mais genéricos (`tabelaPessoas`, `modeloTabela`).
- Usei `PessoaDAO.listar()` como uma abstração do banco de dados.

3. Tipos:

- `real[]` = array de números decimais (`double[]`).
- `inteiro[]` = array de inteiros (`int[]`).
- `matriz[][]` = matriz bidimensional (`Object[][]`).

Essa representação mantém a lógica original enquanto facilita o entendimento em português.