

Adicionando um Botão de Relatório com Matrizes/Vetores em Java

Vou adicionar um botão "Gerar Relatório" que demonstra o uso de matrizes e vetores para processar dados da tabela pessoa. O relatório incluirá:

1. Um resumo estatístico usando vetores
2. Uma matriz com todos os dados para exportação

Modificações necessárias:

1. Primeiro, adicione o botão na TelaPrincipal.java:

Texto = "Relatórios"

Variável = "btnRelatorio"

2. Implemente o método do relatório:

```
private void btnRelatorioActionPerformed(java.awt.event.ActionEvent evt) {  
    List<Pessoa> pessoas = PessoaDAO.listar();  
  
    if (pessoas.isEmpty()) {  
        JOptionPane.showMessageDialog(this, "Nenhum dado para gerar relatório!");  
        return;  
    }  
  
    // 1. USO DE VETORES para cálculos estatísticos  
    double[] salarios = new double[pessoas.size()];  
    int[] idades = new int[pessoas.size()];  
  
    for (int i = 0; i < pessoas.size(); i++) {  
        salarios[i] = pessoas.get(i).getSalario();  
        idades[i] = pessoas.get(i).getIdade();  
    }  
  
    // Cálculos estatísticos  
    double mediaSalarial = calcularMedia(salarios);  
    double maiorSalario = encontrarMaior(salarios);  
    int medialdade = calcularMedia(idades);  
  
    // 2. USO DE MATRIZ para relatório completo  
    // Calcula o tamanho necessário: 5 linhas de cabeçalho + 1 linha de separação + N  
    // linhas de dados  
    int totalLinhas = 5 + 1 + pessoas.size();  
    Object[][] dadosRelatorio = new Object[totalLinhas][4];  
  
    // Cabeçalho  
    dadosRelatorio[0][0] = "RELATÓRIO DE PESSOAS";  
    dadosRelatorio[1][0] = "Total de registros: " + pessoas.size();  
    dadosRelatorio[2][0] = "Média salarial: " + String.format("R$ %.2f", mediaSalarial);  
    dadosRelatorio[2][1] = "Maior salário: " + String.format("R$ %.2f", maiorSalario);  
    dadosRelatorio[3][0] = "Média de idade: " + medialdade + " anos";
```

```

// Linha em branco para separar (linha 4)
dadosRelatorio[4][0] = "--- DETALHES ---";

// Cabeçalho da tabela (linha 5)
dadosRelatorio[5][0] = "ID";
dadosRelatorio[5][1] = "Nome";
dadosRelatorio[5][2] = "Idade";
dadosRelatorio[5][3] = "Salário";

// Dados das pessoas (começando da linha 6)
for (int i = 0; i < pessoas.size(); i++) {
    Pessoa p = pessoas.get(i);
    dadosRelatorio[i + 6][0] = p.getId();
    dadosRelatorio[i + 6][1] = p.getNome();
    dadosRelatorio[i + 6][2] = p.getIdade();
    dadosRelatorio[i + 6][3] = p.getSalario();
}

// Exibir relatório em uma nova janela
exibirRelatorio(dadosRelatorio);
}

```

3. Adicione os métodos auxiliares:

```

private double calcularMedia(double[] valores) {
    double soma = 0;
    for (double valor : valores) {
        soma += valor;
    }
    return soma / valores.length;
}

private int calcularMedia(int[] valores) {
    int soma = 0;
    for (int valor : valores) {
        soma += valor;
    }
    return soma / valores.length;
}

private double encontrarMaior(double[] valores) {
    double maior = valores[0];
    for (double valor : valores) {
        if (valor > maior) {
            maior = valor;
        }
    }
    return maior;
}

```

```

private void exibirRelatorio(Object[][] dados) {
    JFrame frameRelatorio = new JFrame("Relatório Completo");
    JTable tabelaRelatorio = new JTable(dados, new String[]{"", "", "", ""});

    // Desabilita edição
    tabelaRelatorio.setDefaultEditor(Object.class, null);

    // Rolagem
    JScrollPane scrollPane = new JScrollPane(tabelaRelatorio);

    frameRelatorio.add(scrollPane);
    frameRelatorio.setSize(600, 400);
    frameRelatorio.setLocationRelativeTo(null);
    frameRelatorio.setVisible(true);
}

```

ATENÇÃO: Verificar se as importações ao final da implementação:

```

import java.util.List;
import javax.swing.JFrame;
import javax.swing.JOptionPane;
import javax.swing.JScrollPane;
import javax.swing.JTable;
import javax.swing.table.DefaultTableModel;

```

Como ficaria o relatório:

RELATÓRIO DE PESSOAS

Total de registros: 3

Média salarial: R\$ 3166.67 Maior salário: R\$ 4000.00

Média de idade: 30 anos

1	João	25	2500.0
2	Maria	30	3000.0
3	Carlos	35	4000.0

Código com Explicações Detalhadas sobre Vetores e Matrizes

Vou mostrar o trecho do seu código que utiliza vetores e matrizes, com explicações linha por linha em formato comentado:

1. Vetores (Arrays Unidimensionais)

```
// Cria um VETOR para armazenar todos os salários
// double[] = tipo do vetor (números decimais)
// new double[peessoas.size()] = cria um vetor com tamanho igual ao número de pessoas
double[] salarios = new double[peessoas.size()];

// Cria um VETOR para armazenar todas as idades
int[] idades = new int[peessoas.size()];

// Preenche os vetores com dados das pessoas
for (int i = 0; i < pessoas.size(); i++) {
    salarios[i] = pessoas.get(i).getSalario(); // Armazena salário na posição i
    idades[i] = pessoas.get(i).getIdade();    // Armazena idade na posição i
}

// Exemplo: Se tivermos 3 pessoas:
// salarios = [2500.0, 3000.0, 4000.0]
// idades = [25, 30, 35]
```

2. Matriz (Array Bidimensional)

```
// Cria uma MATRIZ para o relatório
// Object[][] = tipo da matriz (pode receber qualquer tipo de dado)
// [totalLinhas][4] = linhas variáveis + 4 colunas fixas (ID, Nome, Idade, Salário)
int totalLinhas = pessoas.size() + 6; // 6 linhas extras para cabeçalhos
Object[][] dadosRelatorio = new Object[totalLinhas][4];

// Preenchendo o CABEÇALHO do relatório
dadosRelatorio[0][0] = "RELATÓRIO DE PESSOAS"; // Linha 0, Coluna 0
dadosRelatorio[1][0] = "Total: " + pessoas.size(); // Linha 1, Coluna 0
dadosRelatorio[2][0] = "Média Salarial: R$" + mediaSalarial; // Linha 2, Coluna 0
dadosRelatorio[3][0] = "Média Idade: " + mediaIdade + " anos"; // Linha 3, Coluna 0

// Linha de SEPARAÇÃO (Linha 4)
dadosRelatorio[4][0] = "--- DADOS ---";

// Cabeçalho da TABELA (Linha 5)
dadosRelatorio[5][0] = "ID";
dadosRelatorio[5][1] = "NOME";
dadosRelatorio[5][2] = "IDADE";
dadosRelatorio[5][3] = "SALÁRIO";

// Preenchendo os DADOS das pessoas (a partir da Linha 6)
for (int i = 0; i < pessoas.size(); i++) {
    Pessoa p = pessoas.get(i);
    dadosRelatorio[i + 6][0] = p.getId();    // Coluna 0 = ID
    dadosRelatorio[i + 6][1] = p.getNome();  // Coluna 1 = Nome
    dadosRelatorio[i + 6][2] = p.getIdade(); // Coluna 2 = Idade
}
```

```
    dadosRelatorio[i + 6][3] = p.getSalario(); // Coluna 3 = Salário
}
```

```
// Exemplo de como fica a matriz com 2 pessoas:
// [ ["RELATÓRIO DE PESSOAS", null, null, null],
//   ["Total: 2", null, null, null],
//   ["Média Salarial: R$2750.0", null, null, null],
//   ["Média Idade: 27 anos", null, null, null],
//   ["--- DADOS ---", null, null, null],
//   ["ID", "NOME", "IDADE", "SALÁRIO"],
//   [1, "João", 25, 2500.0],
//   [2, "Maria", 29, 3000.0] ]
```

3. Métodos Auxiliares com Vetores

```
// Método para calcular MÉDIA usando vetor de números decimais
private double calcularMedia(double[] valores) {
    double soma = 0;
    // Percorre cada elemento do vetor
    for (double valor : valores) {
        soma += valor; // Acumula a soma
    }
    return soma / valores.length; // Divide pelo número de elementos
}
```

```
// Método para encontrar o MAIOR valor em um vetor
private double encontrarMaior(double[] valores) {
    double maior = valores[0]; // Assume que o primeiro é o maior
    // Compara com os demais elementos
    for (double valor : valores) {
        if (valor > maior) {
            maior = valor; // Atualiza se encontrar maior
        }
    }
    return maior;
}
```

4. Exibição da Matriz na Interface

```
// Método para exibir a matriz em uma nova janela
private void exibirRelatorio(Object[][] dados) {
    JFrame frameRelatorio = new JFrame("Relatório Completo");

    // Cria tabela com os dados da matriz
    JTable tabelaRelatorio = new JTable(dados, new String[]{"", "", "", ""});

    // Configurações visuais
    tabelaRelatorio.setDefaultEditor(Object.class, null); // Torna não editável
    JScrollPane scrollPane = new JScrollPane(tabelaRelatorio); // Adiciona rolagem

    frameRelatorio.add(scrollPane);
    frameRelatorio.setSize(600, 400);
}
```

```
frameRelatorio.setLocationRelativeTo(null);
frameRelatorio.setVisible(true);
}
```

DESAFIO: Consegue acrescentar ao final do relatório uma linha com o somatório dos salários?

1) Vou modificar o método **btnRelatorioActionPerformed** para incluir o somatório dos salários no final do relatório:

```
private void btnRelatorioActionPerformed(java.awt.event.ActionEvent evt) {
    List<Pessoa> pessoas = PessoaDAO.listar();

    if (pessoas.isEmpty()) {
        JOptionPane.showMessageDialog(this, "Nenhum dado para gerar relatório!");
        return;
    }

    // 1. USO DE VETORES para cálculos estatísticos
    double[] salarios = new double[pessoas.size()];
    int[] idades = new int[pessoas.size()];

    for (int i = 0; i < pessoas.size(); i++) {
        salarios[i] = pessoas.get(i).getSalario();
        idades[i] = pessoas.get(i).getIdade();
    }

    // Cálculos estatísticos
    double mediaSalarial = calcularMedia(salarios);
    double maiorSalario = encontrarMaior(salarios);
    int medialdade = calcularMedia(idades);
    double somaSalarios = calcularSoma(salarios); // Novo cálculo

    // 2. USO DE MATRIZ para relatório completo
    // Calcula o tamanho necessário:
    // 5 linhas de cabeçalho + 1 linha de separação + N linhas de dados + 1 linha de
total
    int totalLinhas = 5 + 1 + pessoas.size() + 1;
    Object[][] dadosRelatorio = new Object[totalLinhas][4];

    // Cabeçalho
    dadosRelatorio[0][0] = "RELATÓRIO DE PESSOAS";
    dadosRelatorio[1][0] = "Total de registros: " + pessoas.size();
    dadosRelatorio[2][0] = "Média salarial: " + String.format("R$ %.2f", mediaSalarial);
    dadosRelatorio[2][1] = "Maior salário: " + String.format("R$ %.2f", maiorSalario);
    dadosRelatorio[3][0] = "Média de idade: " + medialdade + " anos";

    // Linha em branco para separar (linha 4)
    dadosRelatorio[4][0] = "--- DETALHES ---";
```

```

// Cabeçalho da tabela (linha 5)
dadosRelatorio[5][0] = "ID";
dadosRelatorio[5][1] = "Nome";
dadosRelatorio[5][2] = "Idade";
dadosRelatorio[5][3] = "Salário";

// Dados das pessoas (começando da linha 6)
for (int i = 0; i < pessoas.size(); i++) {
    Pessoa p = pessoas.get(i);
    dadosRelatorio[i + 6][0] = p.getId();
    dadosRelatorio[i + 6][1] = p.getNome();
    dadosRelatorio[i + 6][2] = p.getIdade();
    dadosRelatorio[i + 6][3] = p.getSalario();
}

// Linha de TOTAL (última linha)
int linhaTotal = 6 + pessoas.size();
dadosRelatorio[linhaTotal][0] = "TOTAL:";
dadosRelatorio[linhaTotal][3] = String.format("R$ %.2f",
somaSalarios);

// Exibir relatório em uma nova janela
exibirRelatorio(dadosRelatorio);
}

```

Em seguida vou acrescentar novo método para calcular a soma dos salários:

```

// Novo método para calcular a soma dos salários
private double calcularSoma(double[] valores) {
    double soma = 0;
    for (double valor : valores) {
        soma += valor;
    }
    return soma;
}

```

Explicação das Alterações:

1. Novo cálculo do somatório:
 - Adicionei `double somaSalarios = calcularSoma(salarios);` para calcular o total
 - Criado o método `calcularSoma()` que percorre o vetor de salários e soma todos os valores
2. Ajuste no tamanho da matriz:
 - Aumentei o tamanho da matriz em +1 linha: `int totalLinhas = 5 + 1 + pessoas.size() + 1;`
3. Adição da linha de total:
 - Calcula a posição da última linha: `int linhaTotal = 6 + pessoas.size();`

- Preenche a última linha com:
 - dadosRelatorio[linhaTotal][0] = "TOTAL:" (rótulo)
 - dadosRelatorio[linhaTotal][3] = String.format("R\$ %.2f", somaSalarios) (valor formatado)
- 4. Formatação do valor:
 - Usei String.format("R\$ %.2f", somaSalarios) para mostrar o valor com 2 casas decimais e símbolo de Real
 -

Como ficará o relatório:

RELATÓRIO...

Total: 3

Média...	Maior...
----------	----------

--- DADOS ---

ID	Nome	Idade	Salário
1	João	25	2500.00
2	Maria	30	3000.00
3	Carlos	35	4000.00
TOTAL:			R\$ 9500.00
