

Lista de exercícios

Kevin da Silva Lima 2 DS MAS

1-

```

1  import javax.swing.*;
2  import java.awt.event.*;
3
4  public class Main {
5      public static void main(String[] args) {
6          JFrame frame = new JFrame( title: "Calculadora");
7          frame.setSize( width: 350, height: 300);
8          frame.setLayout(null);
9          frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
10
11         // Campos de entrada
12         JLabel labelN1 = new JLabel( text: "N1");
13         labelN1.setBounds( x: 20, y: 10, width: 20, height: 20);
14         JTextField txtN1 = new JTextField();
15         txtN1.setBounds( x: 50, y: 10, width: 250, height: 20);
16
17         JLabel labelN2 = new JLabel( text: "N2");
18         labelN2.setBounds( x: 20, y: 40, width: 20, height: 20);
19         JTextField txtN2 = new JTextField();
20         txtN2.setBounds( x: 50, y: 40, width: 250, height: 20);
21
22         // Botões de operação
23         // É um botão de opção (tipo "bolinha selecionável")
24         JRadioButton soma = new JRadioButton( text: "Soma (+)");
25         soma.setBounds( x: 30, y: 70, width: 100, height: 20);
26         JRadioButton subtracao = new JRadioButton( text: "Subtração (-)");
27         subtracao.setBounds( x: 30, y: 90, width: 150, height: 20);
28         JRadioButton multiplicacao = new JRadioButton( text: "Multiplicação (*)");
29         multiplicacao.setBounds( x: 30, y: 110, width: 150, height: 20);
30         JRadioButton divisao = new JRadioButton( text: "Divisão (/)");
31         divisao.setBounds( x: 30, y: 130, width: 150, height: 20);
32         JRadioButton resto = new JRadioButton( text: "Resto (%)");
33         resto.setBounds( x: 30, y: 150, width: 150, height: 20);
34
35         // Agrupando botões
36         // Adiciona cada botão ao grupo. Todos passam a se comportar como uma seleção única (um selecionado por vez)
37         ButtonGroup grupo = new ButtonGroup();
38         grupo.add(soma);
39         grupo.add(subtracao);
40         grupo.add(multiplicacao);
41         grupo.add(divisao);
42         grupo.add(resto);


```

```

34
35         // Agrupando botões
36         // Adiciona cada botão ao grupo. Todos passam a se comportar como uma seleção única (um selecionado por vez)
37         ButtonGroup grupo = new ButtonGroup();
38         grupo.add(soma);
39         grupo.add(subtracao);
40         grupo.add(multiplicacao);
41         grupo.add(divisao);
42         grupo.add(resto);
43
44         // Botão calcular
45         JButton btnCalcular = new JButton( text: "Calcular");
46         btnCalcular.setBounds( x: 200, y: 100, width: 100, height: 30);
47
48         // Resultado
49         JLabel labelResultado = new JLabel( text: "Resultado");
50         labelResultado.setBounds( x: 20, y: 190, width: 100, height: 20);
51         JTextField txtResultado = new JTextField();
52         txtResultado.setBounds( x: 100, y: 190, width: 200, height: 20);
53         txtResultado.setEditable(false); // Impede que o usuário digite ou edite o campo - usado apenas para mostrar o resultado
54
55         // Ação do botão calcular
56         btnCalcular.addActionListener(new ActionListener() {
57             public void actionPerformed(ActionEvent e) {
58                 try {
59                     double n1 = Double.parseDouble(txtN1.getText());
60                     double n2 = Double.parseDouble(txtN2.getText());
61                     double resultado = 0;
62
63                     if (soma.isSelected()) {
64                         resultado = n1 + n2;
65                     } else if (subtracao.isSelected()) {
66                         resultado = n1 - n2;
67                     } else if (multiplicacao.isSelected()) {
68                         resultado = n1 * n2;
69                     } else if (divisao.isSelected()) {
70                         resultado = n1 / n2;
71                     } else if (resto.isSelected()) {
72                         resultado = n1 % n2;
73                     }
74

```

```
84         resultado = n1 + n2;
85     } else if (subtracao.isSelected()) {
86         resultado = n1 - n2;
87     } else if (multiplicacao.isSelected()) {
88         resultado = n1 * n2;
89     } else if (divisao.isSelected()) {
90         resultado = n1 / n2;
91     } else if (resto.isSelected()) {
92         resultado = n1 % n2;
93     }
94
95     txtResultado.setText(String.valueOf(resultado)); // Converte o valor numérico para texto e exibe no campo de resultado
96 } catch (NumberFormatException ex) { // Captura erro de conversão quando o usuário digita algo que não é número
97     JOptionPane.showMessageDialog(frame, "Digite números válidos.");
98 }
99
100 }
101
102 // Adicionando componentes ao frame
103 frame.add(labelN1);
104 frame.add(txtN1);
105 frame.add(labelN2);
106 frame.add(txtN2);
107 frame.add(soma);
108 frame.add(subtracao);
109 frame.add(multiplicacao);
110 frame.add(divisao);
111 frame.add(resto);
112 frame.add(btnCalcular);
113 frame.add(labelResultado);
114 frame.add(txtResultado);
115
116 frame.setVisible(true);
117 }
118 }
```

 **Calculadora** — □ ×

N1

N2

☐ Soma (+)

☐ Subtração (-)

☐ Multiplicação (*)

☒ Divisão (/)

☐ Resto (%)

Resultado

2-

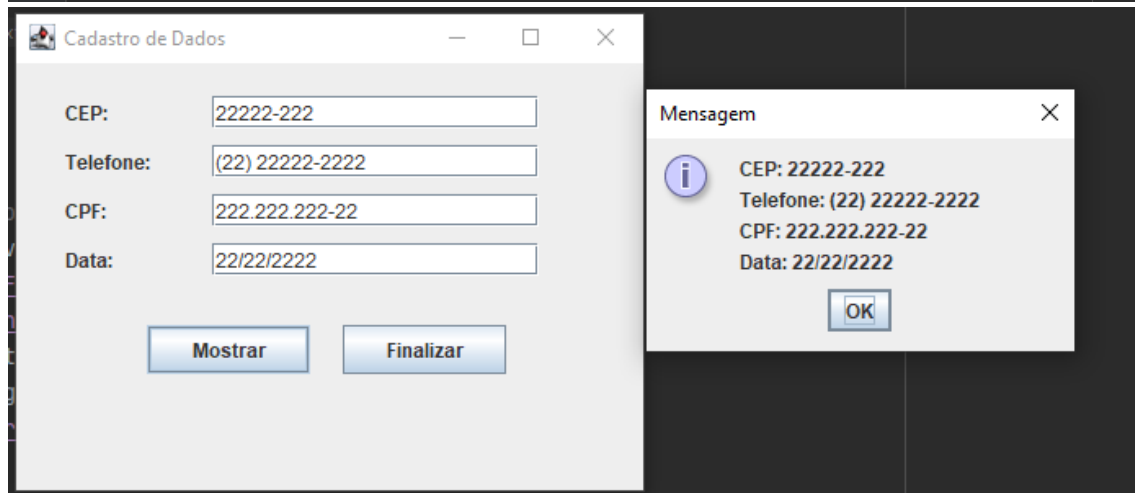
```
1  import javax.swing.*;
2  import javax.swing.text.MaskFormatter;
3  import java.awt.event.*;
4
5  public class Main {
6      public static void main(String[] args) {
7          try {
8
9              // Máscaras de entrada (definem o formato que o usuário deve seguir)
10             // O símbolo '#' indica que apenas números serão aceitos naquela posição
11
12             MaskFormatter cepFormatter = new MaskFormatter("#####-###");
13             MaskFormatter telefoneFormatter = new MaskFormatter("(##) #####-####");
14             MaskFormatter cpfFormatter = new MaskFormatter("###.###.###-##");
15             MaskFormatter dataFormatter = new MaskFormatter("##/##/####");
16
17             // Campos de texto formatados usando as máscaras criadas acima
18             // A palavra-chave 'final' significa que a variável não pode ser reatribuída depois de criada,
19             // mas você ainda pode alterar as propriedades do objeto (como texto, cor, tamanho etc.)
20
21             final JFormattedTextField cepField = new JFormattedTextField(cepFormatter);
22             // Define posição (x, y) e tamanho (largura, altura) do campo na janela
23             cepField.setBounds( x: 120, y: 20, width: 200, height: 20);
24
25             final JFormattedTextField telefoneField = new JFormattedTextField(telefoneFormatter);
26             telefoneField.setBounds( x: 120, y: 50, width: 200, height: 20);
27
28             final JFormattedTextField cpfField = new JFormattedTextField(cpfFormatter);
29             cpfField.setBounds( x: 120, y: 80, width: 200, height: 20);
30
31             final JFormattedTextField dataField = new JFormattedTextField(dataFormatter);
32             dataField.setBounds( x: 120, y: 110, width: 200, height: 20);
33
34             // Criando o frame
35             JFrame frame = new JFrame( title: "Cadastro de Dados");
36             frame.setSize( width: 400, height: 300);
37             frame.setLayout(null);
38             frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
39
40             // Labels
```

```

Main.java x
Download grammar and spelling checker for Portuguese?
40 // Labels
41 JLabel labelCep = new JLabel( text: "CEP:");
42 labelCep.setBounds( x: 30, y: 20, width: 100, height: 20);
43 frame.add(labelCep);
44
45 JLabel labelTelefone = new JLabel( text: "Telefone:");
46 labelTelefone.setBounds( x: 30, y: 50, width: 100, height: 20);
47 frame.add(labelTelefone);
48
49 JLabel labelCpf = new JLabel( text: "CPF:");
50 labelCpf.setBounds( x: 30, y: 80, width: 100, height: 20);
51 frame.add(labelCpf);
52
53 JLabel labelData = new JLabel( text: "Data:");
54 labelData.setBounds( x: 30, y: 110, width: 100, height: 20);
55 frame.add(labelData);
56
57 // Adicionando os campos
58 frame.add(cepField);
59 frame.add(telefoneField);
60 frame.add(cpfField);
61 frame.add(dataField);
62
63 // Botão "Mostrar"
64 JButton botaoMostrar = new JButton( text: "Mostrar");
65 botaoMostrar.setBounds( x: 80, y: 160, width: 100, height: 30);
66 frame.add(botaoMostrar);
67
68 // Botão "Finalizar"
69 JButton botaoFinalizar = new JButton( text: "Finalizar");
70 botaoFinalizar.setBounds( x: 200, y: 160, width: 100, height: 30);
71 frame.add(botaoFinalizar);
72
73 // Ação do botão "Mostrar"
74 botaoMostrar.addActionListener(new ActionListener() {
75     public void actionPerformed(ActionEvent e) {
76         String mensagem = "CEP: " + cepField.getText() +
77             "\nTelefone: " + telefoneField.getText() +
78             "\nCPF: " + cpfField.getText() +
79             "\nData: " + dataField.getText();

```

```
60      frame.add(cpffField);
61      frame.add(dataField);
62
63      // Botão "Mostrar"
64      JButton botaoMostrar = new JButton(text: "Mostrar");
65      botaoMostrar.setBounds(x: 80, y: 160, width: 100, height: 30);
66      frame.add(botaoMostrar);
67
68      // Botão "Finalizar"
69      JButton botaoFinalizar = new JButton(text: "Finalizar");
70      botaoFinalizar.setBounds(x: 200, y: 160, width: 100, height: 30);
71      frame.add(botaoFinalizar);
72
73      // Ação do botão "Mostrar"
74      botaoMostrar.addActionListener(new ActionListener() {
75          public void actionPerformed(ActionEvent e) {
76              String mensagem = "CEP: " + cepField.getText() +
77                  "\nTelefone: " + telefoneField.getText() +
78                  "\nCPF: " + cpfField.getText() +
79                  "\nData: " + dataField.getText();
80              JOptionPane.showMessageDialog(frame, mensagem);
81          }
82      });
83
84      // Ação do botão "Finalizar"
85      botaoFinalizar.addActionListener(e -> System.exit(status: 0));
86
87      frame.setVisible(true);
88
89      } catch (Exception e) { // Captura qualquer erro que ocorra dentro do bloco try
90          e.printStackTrace(); // e imprime detalhes do erro no console para ajudar no diagnóstico
91      }
92  }
93  }
```



3-

```
1  import javax.swing.*;
2  import java.awt.event.ActionEvent;
3  import java.awt.event.ActionListener;
4
5  public class Main {
6      public static void main(String[] args) {
7          JFrame frame = new JFrame("Filmes");
8          frame.setSize( width: 300, height: 300);
9          frame.setLayout(null);
10         frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
11         frame.setLocationRelativeTo(null);
12
13         // Titulo
14         JLabel titulo = new JLabel( text: "Seleção de Filmes");
15         titulo.setBounds( x: 70, y: 10, width: 200, height: 30);
16         // Altera o tamanho da fonte do titulo para 18 pontos, mantendo o estilo atual (negrito, itálico, etc.)
17         titulo.setFont(titulo.getFont().deriveFont( size: 18.0f));
18         frame.add(titulo);
19
20
21         // Checkboxes
22         // Cria uma caixa de seleção com o rótulo "Ação"
23         // O usuário pode marcar ou desmarcar essa opção
24         JCheckBoxacao = new JCheckBox( text: "Ação");
25         JCheckBoxcomedia = new JCheckBox( text: "Comédia");
26         JCheckBoxcomRomantica = new JCheckBox( text: "Comédia Romântica");
27         JCheckBoxficcao = new JCheckBox( text: "Ficção");
28         JCheckBoxromance = new JCheckBox( text: "Romance");
29         JCheckBoxsuspense = new JCheckBox( text: "Suspense");
30         JCheckBoxterror = new JCheckBox( text: "Terror");
31
32        acao.setBounds( x: 30, y: 50, width: 200, height: 20);
33        comedia.setBounds( x: 30, y: 70, width: 200, height: 20);
34        comRomantica.setBounds( x: 30, y: 90, width: 200, height: 20);
35        ficcao.setBounds( x: 30, y: 110, width: 200, height: 20);
36        romance.setBounds( x: 30, y: 130, width: 200, height: 20);
37        suspense.setBounds( x: 30, y: 150, width: 200, height: 20);
38        terror.setBounds( x: 30, y: 170, width: 200, height: 20);
39
40         ButtonGroup grupo = new ButtonGroup();
41         grupo.add(acao);
42         grupo.add(comedia);
43         grupo.add(comRomantica);
44         grupo.add(ficcao);
45         grupo.add(romance);
46         grupo.add(suspense);
47         grupo.add(terror);
48
49
50         // Adiciona ao frame
51         frame.add(acao);
52         frame.add(comedia);
53         frame.add(comRomantica);
54         frame.add(ficcao);
55         frame.add(romance);
56         frame.add(suspense);
57         frame.add(terror);
58     }
59 }
```

```

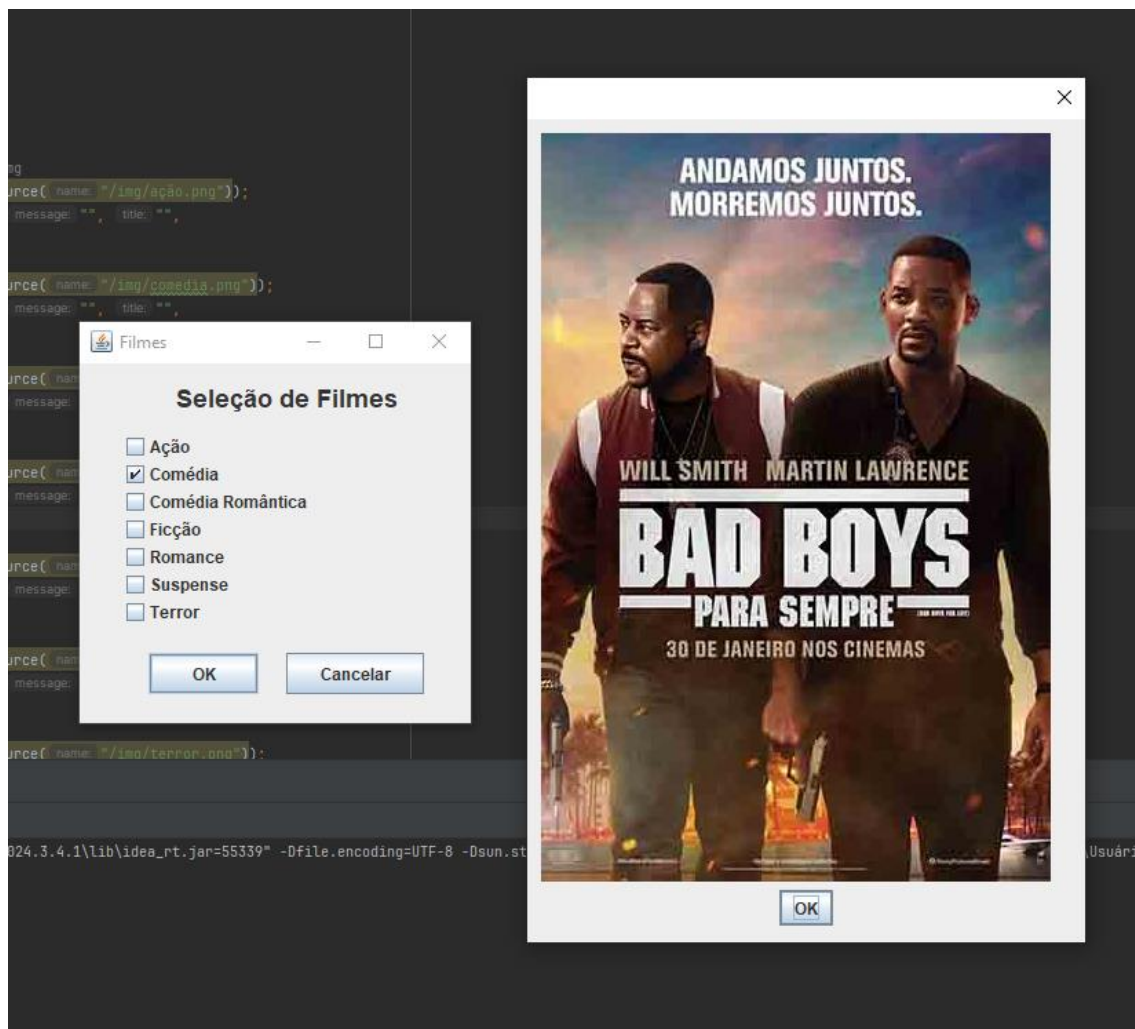
Main.java x
5 public class Main {
6     public static void main(String[] args) {
7         // Adicionando ao frame
8
9         frame.add(acao);
10        frame.add(comedia);
11        frame.add(comRomantica);
12        frame.add(ficcao);
13        frame.add(romance);
14        frame.add(suspense);
15        frame.add(terror);
16
17        // Botões
18        JButton ok = new JButton( "OK");
19        ok.setBounds( x: 50, y: 210, width: 80, height: 30);
20        JButton cancelar = new JButton( "Cancelar");
21        cancelar.setBounds( x: 150, y: 210, width: 100, height: 30);
22
23        ok.addActionListener(new ActionListener() {
24            @Override
25            public void actionPerformed(ActionEvent e) {
26
27                if(acao.isSelected()){
28                    // Mostra a imagem "ação.png" dentro da pasta src/img
29                    ImageIcon imagem = new ImageIcon(Main.class.getResource( name: "/img/ação.png"));
30                    JOptionPane.showMessageDialog( parentComponent: null, message: "", title: "",
31                        JOptionPane.INFORMATION_MESSAGE, imagem);
32                } else if (comedia.isSelected()) {
33                    ImageIcon imagem = new ImageIcon(Main.class.getResource( name: "/img/comedia.png"));
34                    JOptionPane.showMessageDialog( parentComponent: null, message: "", title: "",
35                        JOptionPane.INFORMATION_MESSAGE, imagem);
36                }else if (comRomantica.isSelected()) {
37                    ImageIcon imagem = new ImageIcon(Main.class.getResource( name: "/img/comedia romantica.png"));
38                    JOptionPane.showMessageDialog( parentComponent: null, message: "", title: "",
39                        JOptionPane.INFORMATION_MESSAGE, imagem);
40                }else if (ficcao.isSelected()) {
41                    ImageIcon imagem = new ImageIcon(Main.class.getResource( name: "/img/ficção.png"));
42                    JOptionPane.showMessageDialog( parentComponent: null, message: "", title: "",
43                        JOptionPane.INFORMATION_MESSAGE, imagem);
44                }else if (romance.isSelected()) {
45                    ImageIcon imagem = new ImageIcon(Main.class.getResource( name: "/img/romance.png"));
46                    JOptionPane.showMessageDialog( parentComponent: null, message: "", title: "",
47                        JOptionPane.INFORMATION_MESSAGE, imagem);
48                }else if (suspense.isSelected()) {
49                    ImageIcon imagem = new ImageIcon(Main.class.getResource( name: "/img/suspense.png"));
50                    JOptionPane.showMessageDialog( parentComponent: null, message: "", title: "",
51                        JOptionPane.INFORMATION_MESSAGE, imagem);
52                }else if (terror.isSelected()) {
53                    ImageIcon imagem = new ImageIcon(Main.class.getResource( name: "/img/terror.png"));
54                    JOptionPane.showMessageDialog( parentComponent: null, message: "", title: "",
55                        JOptionPane.INFORMATION_MESSAGE, imagem);
56                }
57            }
58        });
59
60        cancelar.addActionListener( ActionEvent e -> System.exit( status: 0));
61    }
62}
```



```

Main.java x
5     public class Main {
6         public static void main(String[] args) {
7             // Criação da janela
8             JFrame frame = new JFrame("Janela de Teste");
9             frame.setSize(400, 300);
10            frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
11
12            JButton cancelar = new JButton("Cancelar");
13            cancelar.setBounds(150, 210, 100, 30);
14
15            ok.addActionListener(new ActionListener() {
16                @Override
17                public void actionPerformed(ActionEvent e) {
18
19                    if(acao.isSelected()){
20                        // Mostra a imagem "ação.png" dentro da pasta src/img
21                        ImageIcon imagem = new ImageIcon(Main.class.getResource("/img/ação.png"));
22                        JOptionPane.showMessageDialog(parentComponent: null, message: "", title: "",
23                            JOptionPane.INFORMATION_MESSAGE, imagem);
24                    } else if (comedia.isSelected()) {
25                        ImageIcon imagem = new ImageIcon(Main.class.getResource("/img/comedia.png"));
26                        JOptionPane.showMessageDialog(parentComponent: null, message: "", title: "",
27                            JOptionPane.INFORMATION_MESSAGE, imagem);
28                    } else if (comRomantica.isSelected()) {
29                        ImageIcon imagem = new ImageIcon(Main.class.getResource("/img/comedia_romantica.png"));
30                        JOptionPane.showMessageDialog(parentComponent: null, message: "", title: "",
31                            JOptionPane.INFORMATION_MESSAGE, imagem);
32                    } else if (ficcao.isSelected()) {
33                        ImageIcon imagem = new ImageIcon(Main.class.getResource("/img/ficção.png"));
34                        JOptionPane.showMessageDialog(parentComponent: null, message: "", title: "",
35                            JOptionPane.INFORMATION_MESSAGE, imagem);
36                    } else if (romance.isSelected()) {
37                        ImageIcon imagem = new ImageIcon(Main.class.getResource("/img/romance.png"));
38                        JOptionPane.showMessageDialog(parentComponent: null, message: "", title: "",
39                            JOptionPane.INFORMATION_MESSAGE, imagem);
40                    } else if (suspense.isSelected()) {
41                        ImageIcon imagem = new ImageIcon(Main.class.getResource("/img/suspense.png"));
42                        JOptionPane.showMessageDialog(parentComponent: null, message: "", title: "",
43                            JOptionPane.INFORMATION_MESSAGE, imagem);
44                    } else if (terror.isSelected()) {
45                        ImageIcon imagem = new ImageIcon(Main.class.getResource("/img/terror.png"));
46                        JOptionPane.showMessageDialog(parentComponent: null, message: "", title: "",
47                            JOptionPane.INFORMATION_MESSAGE, imagem);
48                    }
49                }
50            });
51
52            cancelar.addActionListener( ActionEvent e -> System.exit( status: 0));
53
54            frame.add(ok);
55            frame.add(cancelar);
56
57            frame.setVisible(true);
58        }
59    }

```



4-

Main.java

```
1  import javax.swing.*;
2  import java.awt.*;
3  import java.awt.event.ActionEvent;
4  import java.awt.event.ActionListener;
5
6  public class Main extends JFrame {
7      JFormattedTextField texNome, texClasse, textRm; 4 usages
8
9      public Main() { 1 usage
10         super( title: "Empréstimo de Livro");
11         setLayout(null);
12         setSize( width: 450, height: 300);
13         setResizable(false);
14         setLocationRelativeTo(null);
15         setDefaultCloseOperation(EXIT_ON_CLOSE);
16
17         // Titulo
18         JLabel titulo = new JLabel( text: "Empréstimo");
19         titulo.setBounds( x: 170, y: 10, width: 200, height: 30);
20         titulo.setFont(titulo.getFont().deriveFont( size: 16.0f));
21         titulo.setForeground(Color.RED);
22         add(titulo); // Adiciona o titulo à janela
23
24         // ComboBox de livros
25         String[] livros = {"Livro1", "Livro2", "Livro3", "Livro4"};
26         JComboBox<String> comboLivros = new JComboBox<>(livros);
27         comboLivros.setBounds( x: 300, y: 50, width: 100, height: 20);
28         add(comboLivros);
29
30         // Nome
31         JLabel lblNome = new JLabel( text: "Nome:");
32         lblNome.setBounds( x: 20, y: 50, width: 50, height: 20);
33         add(lblNome);
34
35         texNome = new JFormattedTextField();
36         texNome.setBounds( x: 70, y: 50, width: 200, height: 20);
37         add(texNome);
38
39         // Classe
40         JLabel lblClasse = new JLabel( text: "Classe:");
41         lblClasse.setBounds( x: 20, y: 90, width: 50, height: 20);
42         add(lblClasse);
43
44         texClasse = new JFormattedTextField();
45         texClasse.setBounds( x: 70, y: 90, width: 200, height: 20);
46         add(texClasse);
47
48         // RM
49         JLabel lblRM = new JLabel( text: "RM:");
50         lblRM.setBounds( x: 20, y: 130, width: 50, height: 20);
51         add(lblRM);
52
53         textRm = new JFormattedTextField();
54         textRm.setBounds( x: 70, y: 130, width: 200, height: 20);
55         add(textRm);
56     }
```

```
1 public class Main extends JFrame {
2     public Main() {
3         setTitle("Empréstimo de Livro");
4         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
5         setSize(400, 300);
6
7         // Classe
8         JLabel lblClasse = new JLabel("Classe:");
9         lblClasse.setBounds(x: 20, y: 90, width: 50, height: 20);
10        add(lblClasse);
11
12        texClasse = new JFormattedTextField();
13        texClasse.setBounds(x: 70, y: 90, width: 200, height: 20);
14        add(texClasse);
15
16        // RM
17        JLabel lblRM = new JLabel("RM:");
18        lblRM.setBounds(x: 20, y: 130, width: 50, height: 20);
19        add(lblRM);
20
21        textRm = new JFormattedTextField();
22        textRm.setBounds(x: 70, y: 130, width: 200, height: 20);
23        add(textRm);
24
25        // Botão Empréstimo
26        JButton btnEmpréstimo = new JButton("Empréstimo");
27        btnEmpréstimo.setBounds(x: 160, y: 180, width: 120, height: 30);
28        add(btnEmpréstimo);
29
30        // Ação do botão
31        btnEmpréstimo.addActionListener(new ActionListener() {
32            @Override
33            public void actionPerformed(ActionEvent e) {
34                // Pega o texto digitado no campo 'texNome' e remove espaços em branco no início e no fim
35                String nome = texNome.getText().trim(); // trim() - remove espaços extras no começo e no fim da string
36                String classe = texClasse.getText().trim();
37                String rm = textRm.getText().trim();
38                String livro = (String) comboLivros.getSelectedItem();
39
40                // Exibir as informações em um JOptionPane
41                JOptionPane.showMessageDialog(parentComponent: null,
42                    message: "Nome: " + nome +
43                        "\nClasse: " + classe +
44                        "\nRM: " + rm +
45                        "\nLivro: " + livro,
46                    title: "Empréstimo Realizado", // Título da janela
47                    JOptionPane.INFORMATION_MESSAGE);
48            }
49        });
50
51        setVisible(true);
52    }
53
54    public static void main(String[] args) {
55        new Main();
56    }
57 }
```

