

Algoritmos III

Swing: JTextField, JFormattedTextField, JLabel e JComboBox

Raphael de Souza, Karen Figueiredo

IC/UFMT

1 Swing: JTextField, JFormattedTextField, JLabel e JComboBox

- javax.swing.JLabel
- javax.swing.JTextField
- javax.swing.JComboBox
- javax.swing.JFormattedTextField

Swing:Componentes

- JTextField
- JFormattedTextField
- JLabel
- JRadioButton
- JComboBox
- JTextArea
- JPasswordField
- JTree
- JTable
- JMenuBar
- JMenuItem

Swing:Componentes

- **JTextField**
- JFormattedTextField
- JLabel
- JRadioButton
- JComboBox
- JTextArea
- JPasswordField
- JTree
- JTable
- JMenuBar
- JMenuItem

Swing:Componentes

- JTextField
- JFormattedTextField
- JLabel
- JRadioButton
- JComboBox
- JTextArea
- JPasswordField
- JTree
- JTable
- JMenuBar
- JMenuItem

Swing:Componentes

- **JTextField**
- **JFormattedTextField**
- **JLabel**
- JRadioButton
- JComboBox
- JTextArea
- JPasswordField
- JTree
- JTable
- JMenuBar
- JMenuItem

Swing:Componentes

- JTextField
- JFormattedTextField
- JLabel
- JRadioButton
- JComboBox
- JTextArea
- JPasswordField
- JTree
- JTable
- JMenuBar
- JMenuItem

Swing:Componentes

- JTextField
- JFormattedTextField
- JLabel
- JRadioButton
- JComboBox
- JTextArea
- JPasswordField
- JTree
- JTable
- JMenuBar
- JMenuItem

Swing:Componentes

- JTextField
- JFormattedTextField
- JLabel
- JRadioButton
- JComboBox
- JTextArea
- JPasswordField
- JTree
- JTable
- JMenuBar
- JMenuItem

Swing:Componentes

- JTextField
- JFormattedTextField
- JLabel
- JRadioButton
- JComboBox
- JTextArea
- JPasswordField
- JTree
- JTable
- JMenuBar
- JMenuItem

Swing:Componentes

- JTextField
- JFormattedTextField
- JLabel
- JRadioButton
- JComboBox
- JTextArea
- JPasswordField
- JTree
- JTable
- JMenuBar
- JMenuItem

Swing:Componentes

- JTextField
- JFormattedTextField
- JLabel
- JRadioButton
- JComboBox
- JTextArea
- JPasswordField
- JTree
- JTable
- JMenuBar
- JMenuItem

Swing:Componentes

- JTextField
- JFormattedTextField
- JLabel
- JRadioButton
- JComboBox
- JTextArea
- JPasswordField
- JTree
- JTable
- JMenuBar
- JMenuItem

Swing:Componentes

- JTextField
- JFormattedTextField
- JLabel
- JRadioButton
- JComboBox
- JTextArea
- JPasswordField
- JTree
- JTable
- JMenuBar
- JMenuItem

javax.swing.JLabel

Definição

Componente que serve como texto estático

Definição

Componente que serve como texto estático

javax.swing.JLabel

```
import javax.swing.JLabel;
import javax.swing.SwingConstants;
...
public class Tela extends javax.swing.JFrame {
    public Tela() {
        ...
        JLabel texto = new JLabel();
        texto.setSize(200, 25);
        texto.setLocation(10, 10);
        texto.setText("Texto:");
        texto.setHorizontalAlignment(SwingConstants.
            RIGHT);
        texto.setToolTipText("Texto tool tip");
        add(texto);
    }
}
```

javax.swing.JLabel



javax.swing.JTextField

Definição

Componente que serve como campo de texto

javax.swing.JTextField

Definição

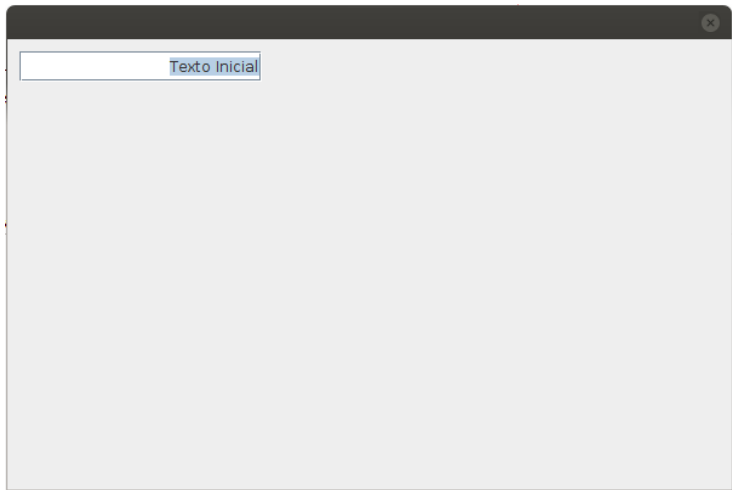
Componente que serve como campo de texto

javax.swing.JTextField

```
import javax.swing.JTextField;
...
public class Tela extends javax.swing.JFrame {
    public Tela() {
        ...

        JTextField texto = new JTextField();
        texto.setSize(200, 25);
        texto.setLocation(10, 10);
        texto.setText("Texto Inicial");
        texto.selectAll();
        texto.setHorizontalAlignment(SwingConstants.
            RIGHT);
        texto.setToolTipText("Texto tool tip");
        add(texto);
    }
}
```

javax.swing.JTextField



javax.swing.JComboBox

Definição

Componente que serve como uma lista pré-determinada

Definição

Componente que serve como uma lista pré-determinada

javax.swing.JComboBox

```
import javax.swing.JLabel;  
import javax.swing.SwingConstants;  
...  
public class Tela extends javax.swing.JFrame {  
    public Tela() {  
        ...  
        JComboBox combo = new JComboBox();  
        combo.addItem("Selezione");  
        combo.addItem("Objecto1");  
        combo.addItem("Objecto2");  
        add(combo);  
    }  
}
```

javax.swing.JComboBox



javax.swing.JFormattedTextField

Definição

Componente que serve como campo de texto formatado

javax.swing.JFormattedTextField

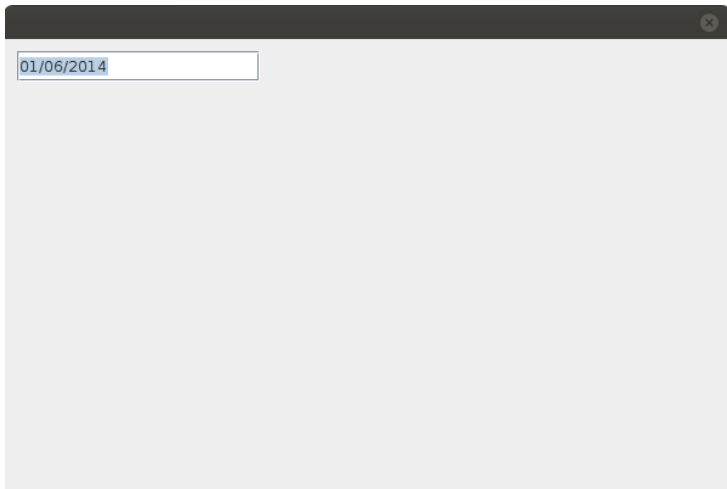
Definição

Componente que serve como campo de texto formatado

javax.swing.JFormattedTextField

```
import javax.swing.JFormattedTextField;
...
try {
    JFormattedTextField texto = new
        JFormattedTextField();
    MaskFormatter mask = new MaskFormatter("//");
    DefaultFormatterFactory dff = new
        DefaultFormatterFactory(mask);
    texto.setFormatterFactory(dff);
    texto.setSize(200, 25);
    texto.setLocation(10, 10);
    texto.setText("01/06/2014");
    texto.selectAll();
    add(texto);
} catch (ParseException ex) { }
```

javax.swing.JFormattedTextField



javax.swing.text.MaskFormatter

Para criar uma máscara pode-se utilizar os seguintes caracteres especiais:

- **#**: Aceita qualquer dígito
- **U**: Aceita qualquer literal e o coloca em maiúsculo
- **L**: Aceita qualquer literal e o coloca em minúsculo
- **A**: Aceita qualquer alfa-numérico
- **'**: Utilizado como carácter escape
- **?**: Qualquer carácter literal
- *****: Qualquer tecla
- **H**: Chave hexadecimal

javax.swing.text.MaskFormatter

Para criar uma máscara pode-se utilizar os seguintes caracteres especiais:

- **#**: Aceita qualquer dígito
- **U**: Aceita qualquer literal e o coloca em maiúsculo
- **L**: Aceita qualquer literal e o coloca em minúsculo
- **A**: Aceita qualquer alfa-numérico
- **'**: Utilizado como carácter escape
- **?**: Qualquer carácter literal
- *****: Qualquer tecla
- **H**: Chave hexadecimal

javax.swing.text.MaskFormatter

Para criar uma máscara pode-se utilizar os seguintes caracteres especiais:

- **#**: Aceita qualquer dígito
- **U**: Aceita qualquer literal e o coloca em maiúsculo
- **L**: Aceita qualquer literal e o coloca em minúsculo
- **A**: Aceita qualquer alfa-numérico
- **'**: Utilizado como carácter escape
- **?**: Qualquer carácter literal
- *****: Qualquer tecla
- **H**: Chave hexadecimal

javax.swing.text.MaskFormatter

Para criar uma máscara pode-se utilizar os seguintes caracteres especiais:

- **#**: Aceita qualquer dígito
- **U**: Aceita qualquer literal e o coloca em maiúsculo
- **L**: Aceita qualquer literal e o coloca em minúsculo
- **A**: Aceita qualquer alfa-numérico
- **'**: Utilizado como carácter escape
- **?**: Qualquer carácter literal
- *****: Qualquer tecla
- **H**: Chave hexadecimal

javax.swing.text.MaskFormatter

Para criar uma máscara pode-se utilizar os seguintes caracteres especiais:

- **#**: Aceita qualquer dígito
- **U**: Aceita qualquer literal e o coloca em maiúsculo
- **L**: Aceita qualquer literal e o coloca em minúsculo
- **A**: Aceita qualquer alfa-numérico
- **'**: Utilizado como carácter escape
- **?**: Qualquer carácter literal
- *****: Qualquer tecla
- **H**: Chave hexadecimal

javax.swing.text.MaskFormatter

Para criar uma máscara pode-se utilizar os seguintes caracteres especiais:

- **#**: Aceita qualquer dígito
- **U**: Aceita qualquer literal e o coloca em maiúsculo
- **L**: Aceita qualquer literal e o coloca em minúsculo
- **A**: Aceita qualquer alfa-numérico
- **'**: Utilizado como carácter escape
- **?**: Qualquer carácter literal
- *****: Qualquer tecla
- **H**: Chave hexadecimal

javax.swing.text.MaskFormatter

Para criar uma máscara pode-se utilizar os seguintes caracteres especiais:

- **#**: Aceita qualquer dígito
- **U**: Aceita qualquer literal e o coloca em maiúsculo
- **L**: Aceita qualquer literal e o coloca em minúsculo
- **A**: Aceita qualquer alfa-numérico
- **'**: Utilizado como carácter escape
- **?**: Qualquer carácter literal
- *****: Qualquer tecla
- **H**: Chave hexadecimal

javax.swing.text.MaskFormatter

Para criar uma máscara pode-se utilizar os seguintes caracteres especiais:

- **#**: Aceita qualquer dígito
- **U**: Aceita qualquer literal e o coloca em maiúsculo
- **L**: Aceita qualquer literal e o coloca em minúsculo
- **A**: Aceita qualquer alfa-numérico
- **'**: Utilizado como carácter escape
- **?**: Qualquer carácter literal
- *****: Qualquer tecla
- **H**: Chave hexadecimal

Para criar uma máscara pode-se utilizar os seguintes caracteres especiais:

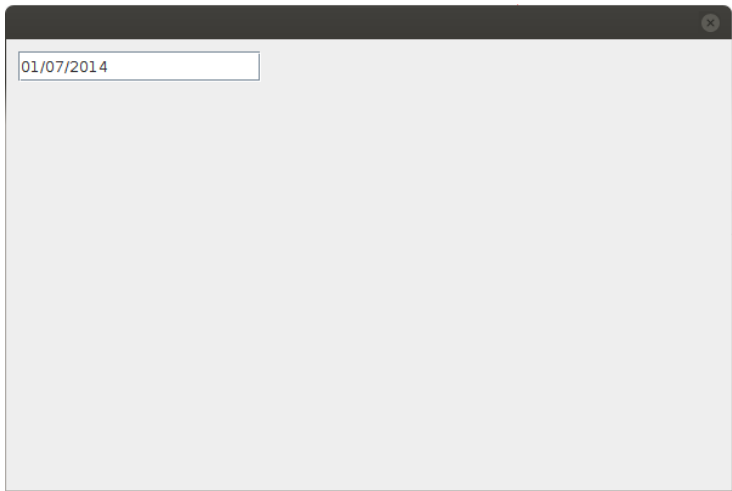
- **#**: Aceita qualquer dígito
- **U**: Aceita qualquer literal e o coloca em maiúsculo
- **L**: Aceita qualquer literal e o coloca em minúsculo
- **A**: Aceita qualquer alfa-numérico
- **'**: Utilizado como carácter escape
- **?**: Qualquer carácter literal
- *****: Qualquer tecla
- **H**: Chave hexadecimal

javax.swing.JFormattedTextField

```
import javax.swing.JFormattedTextField;
...

JFormattedTextField texto = new
    JFormattedTextField();
texto.setSize(200, 25);
texto.setLocation(10, 10);
texto.setValue(new java.util.Date());
add(texto);
```


javax.swing.JFormattedTextField



javax.swing.JFormattedTextField

As classes que já possuem formato padrão são:

- java.util.Date
- java.lang.Number
- java.text.Format
- java.text.NumberFormat
- java.text.DateFormat

javax.swing.JFormattedTextField

As classes que já possuem formato padrão são:

- java.util.Date
- java.lang.Number
- java.text.Format
- java.text.NumberFormat
- java.text.DateFormat

javax.swing.JFormattedTextField

As classes que já possuem formato padrão são:

- java.util.Date
- java.lang.Number
- java.text.Format
- java.text.NumberFormat
- java.text.DateFormat

javax.swing.JFormattedTextField

As classes que já possuem formato padrão são:

- java.util.Date
- java.lang.Number
- java.text.Format
- java.text.NumberFormat
- java.text.DateFormat

javax.swing.JFormattedTextField

As classes que já possuem formato padrão são:

- java.util.Date
- java.lang.Number
- java.text.Format
- java.text.NumberFormat
- java.text.DateFormat

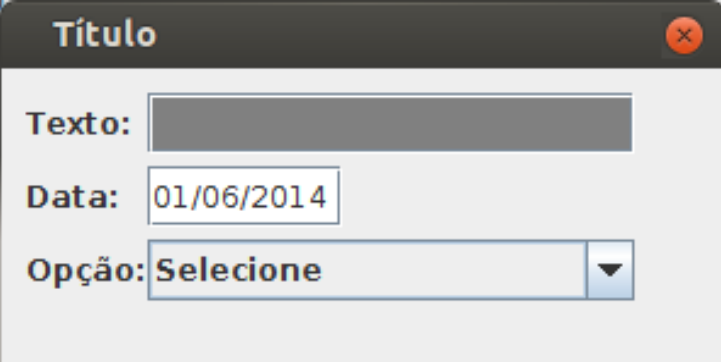
javax.swing.JFormattedTextField

As classes que já possuem formato padrão são:

- java.util.Date
- java.lang.Number
- java.text.Format
- java.text.NumberFormat
- java.text.DateFormat

Prática

Construamos a seguinte tela:



Título

Texto:

Data:

Opção: ▼

Obs: Coloque pelo menos três opções o combo

Referências Bibliográficas

Aula baseada em slides preparados pelos seguintes professores:

- Prof. Bruno B. Boniati - www.cafw.ufsm.br/bruno - Universidade Federal de Santa Maria
- Prof. José Gustavo de Souza Paiva - <http://www.facom.ufu.br/jgustavo/disc/poo.html> - Faculdade de Computação da Universidade Federal de Uberlândia
- Prof. Vítor E. Silva Souza - <http://www.inf.ufes.br/vitorsouza/pt/> - Departamento de Informática da Universidade Federal do Espírito Santo

Livro:

- Guia de Estudo: Certificação Sun para Programador Java 6. Kathy Sierra e Bert Bates