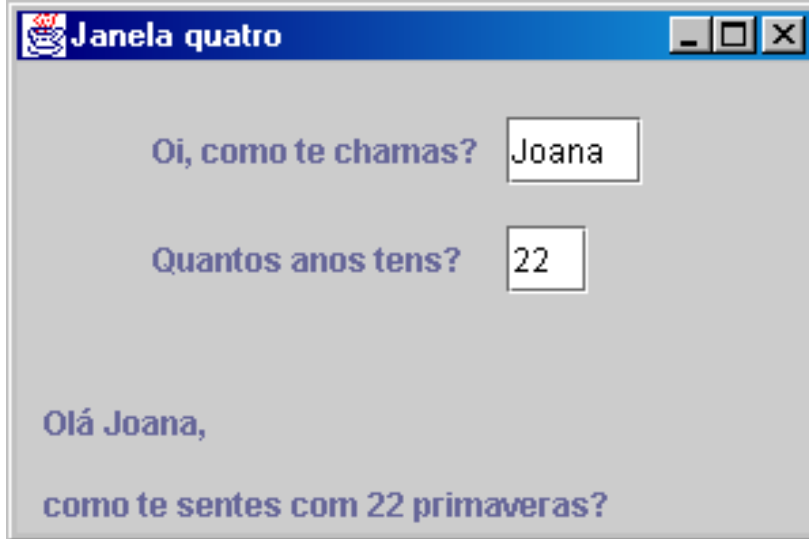


A classe JTextField

- Um campo de texto, permite ao utilizador introduzir uma única linha de texto
- Uma instância da classe JTextField gera uma instância da classe `ActionEvent` quando o objecto está activo e o utilizador pressiona a tecla ENTER.

Suponhamos que queremos construir a janela abaixo,



tratando os eventos gerados quando o utilizador pressiona a tecla ENTER em cada um dos campos de texto.

<http://www.di.ubi.pt/~pprata/poo.htm>

```
public class Janela4 extends JFrame implements ActionListener {
    ...
private JLabel lNome, lIdade, lNome2, lIdade2;
private JTextField nome, idade;
private Container contentor;

public Janela4() {
    ...
contentor = getContentPane();
contentor.setLayout(null);
// criar a etiqueta com a 1º pergunta
lNome = new JLabel("Oi, como te chamas? ");
lNome.setBounds(50,20,125,25);
contentor.add(lNome);
// criar o campo de texto para o nome
nome = new JTextField();
nome.setBounds(180,20, 50,25 );
contentor.add(nome);
// registar o “listener” no campo de texto
nome.addActionListener(this);
// repetir para a 2ª pergunta (atenção às coordenadas)
... }
```

<http://www.di.ubi.pt/~pprata/poo.htm>

```
public void actionPerformed(ActionEvent evt) {
    JTextField campoTexto = (JTextField) evt.getSource();
    int w = campoTexto.getWidth();
    String texto = campoTexto.getText();
    if (w==50)
        mostraNome(texto);
    else
        mostraIdade(texto);
}

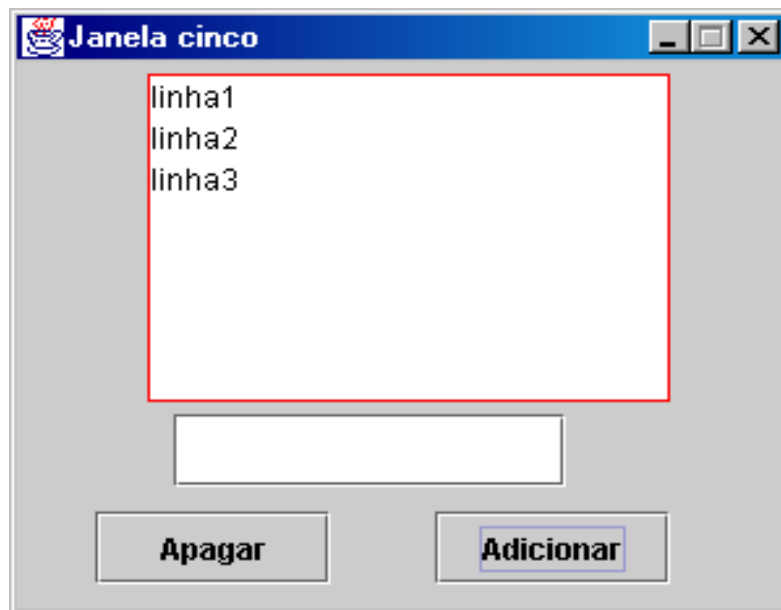
public void mostraNome(String n) {
    INome2 = new JLabel("Olá " + n + ",");
    INome2.setBounds(10,120,100,25);
    contentor.add(INome2);
}

public void mostraIdade(String n) {
    int i = Integer.parseInt(n);
    IIdade2 =
        new JLabel("como te sentes com " + i + " primaveras?");
    IIdade2.setBounds(10,150,250,25);
    contentor.add(IIdade2);
}
```

A classe JTextArea

- Uma área de texto permite introduzir várias linhas de texto

Suponhamos a seguinte janela:



Cada linha de texto introduzida no campo de texto será adicionada à área de texto sempre que o utilizador pressionar a tecla RETURN ou o botão Adicionar. Quando pressionado o botão Apagar será eliminado todo o conteúdo da área de texto.

<http://www.di.ubi.pt/~pprata/poo.htm>

```
private JButton botaoApagar;  
private JButton botaoAdicionar;  
private JTextField linha;  
private JTextArea texto;  
  
// no construtor criar a área de texto  
  
texto = new JTextArea();  
texto.setBounds(50,5,200, 135);  
  
// por omissão a área de texto não é delimitada  
texto.setBorder(BorderFactory.createLineBorder (Color.red));  
texto.setEditable(false);  
contentor.add(texto);
```

Para adicionar texto à área de texto usamos o método
void append (String) da classe JTextArea

Para substituir o texto da área de texto usamos o método
void setText (String) da classe JTextArea

Tratar os eventos

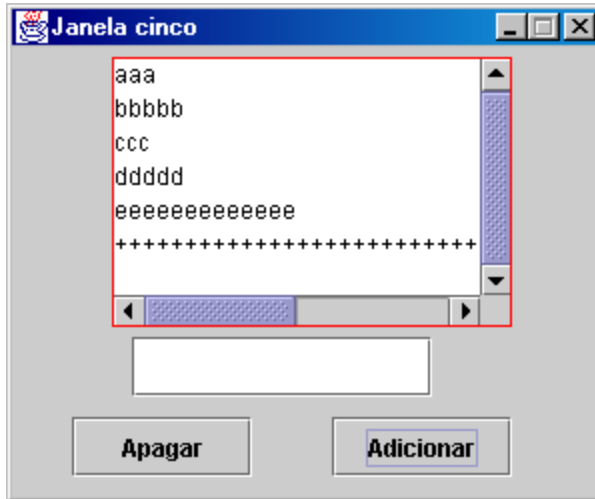
```
public void actionPerformed(ActionEvent evt) {
    if ( evt.getSource() instanceof JButton ) {
        JButton botao = (JButton) evt.getSource();
        if ( botao == botaoAdicionar ) {
            adicionarTexto ( linha.getText() );
        } else {
            apagarTexto();
        }
    } else { // event source: linha
        adicionarTexto(linha.getText() );
    }
}

private void adicionarTexto(String s) {
    texto.append( s + "\n");
    linha.setText("");
}

private void apagarTexto() {
    texto.setText("");
    linha.setText("");
}
```

Scroll bars

Para permitir a visualização de um texto que exceda as margens,



modificamos o código anterior onde criámos a área de texto englobando esta num objecto do tipo “JScrollPane”:

```
texto = new JTextArea();
texto.setEditable(false);
JScrollPane elevador = new JScrollPane (texto);
elevador.setBounds (50,5,200, 135);
elevador.setBorder (
    BorderFactory.createLineBorder (Color.red));
contentor.add(elevador);
```

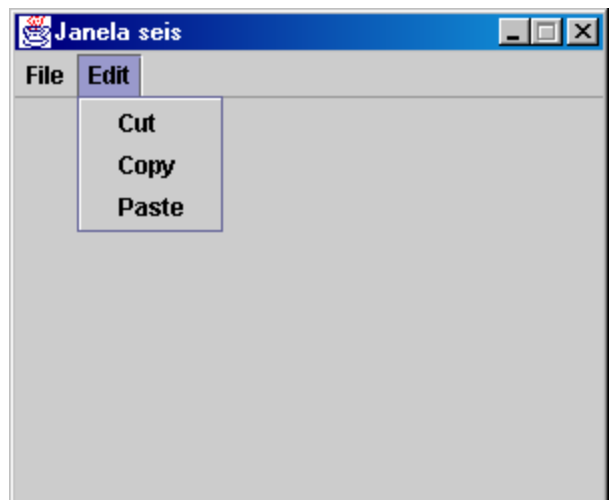
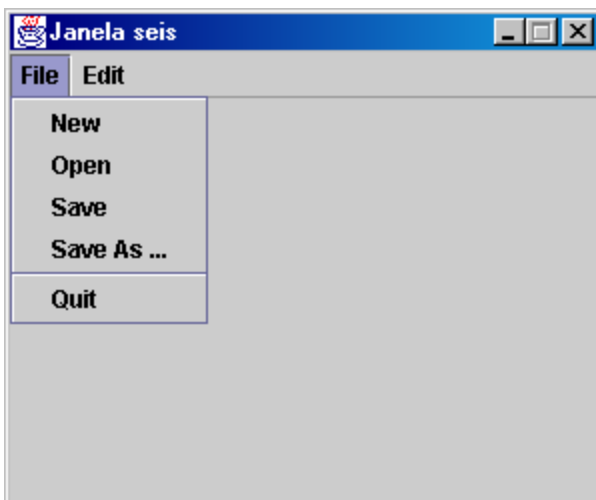
As classes JMenuBar, JMenu, JMenuItem

- . Uma barra de menus (JMenuBar) é a barra onde são colocados os menus
- . Um menu (JMenu) é um item de uma barra de menus
- . Um item de menu (JMenuItem) é uma das opções de um menu

Quando um item de um menu é seleccionado, é gerado um “action event”

Para processar a selecção de um item de um menu associamos um “listener” a esse item.

Vamos construir uma janela com dois menus:



<http://www.di.ubi.pt/~pprata/poo.htm>

Sequência de passos para criar os menus:

- 1 – criar uma barra de menus e adicionar à janela
- 2 – criar os menus
- 3 – criar os items de menu e adicionar ao menu correspondente
- 4 – adicionar cada menu à barra de menus

```
public class Janela6 extends JFrame implements ActionListener {
...
private JLabel resposta;
private JMenu fileMenu, editMenu;
public Janela6() {
...
// criar a barra de menus
JMenuBar barraMenus = new JMenuBar();
//criar dois menus e os seus items
criarFileMenu();
criarEditMenu();
//colocar a barra de menus na janela
setJMenuBar ( barraMenus);
// adicionar os menus à barra de menus
barraMenus.add(fileMenu);
barraMenus.add(editMenu);
```

<http://www.di.ubi.pt/~pprata/poo.htm>

```
public void criarFileMenu() {  
JMenuItem item;  
fileMenu = new JMenu("File");  
  
item = new JMenuItem("New");  
item.addActionListener(this);  
fileMenu.add(item);  
  
item = new JMenuItem("Open");  
item.addActionListener(this);  
fileMenu.add(item);  
  
item = new JMenuItem("Save");  
item.addActionListener(this);  
fileMenu.add(item);  
...  
fileMenu.addSeparator(); // linha horizontal  
  
item = new JMenuItem("Quit");  
item.addActionListener(this);  
fileMenu.add(item);  
}
```

<http://www.di.ubi.pt/~pprata/poo.htm>

// análogo para o método criarEditMenu

Tratar a selecção de um item

```
public void actionPerformed(ActionEvent evt){
    String itemSeleccionado;
    itemSeleccionado = evt.getActionCommand();
    if ( itemSeleccionado.equals("Quit") ) {
        System.exit(0);
    } else {
        resposta.setText("O item seleccionado foi " +
            itemSeleccionado );
    }
}
```

