

Objectivos:

. Construir programas com interfaces gráficas

“Graphical User Interface (GUI) application programs”

- Utilizar as classes **JFrame**, **JButton**, **JLabel**, **ImageIcon**, **JTextField**, **JTextArea** e **JMenu** do package **javax.swing**

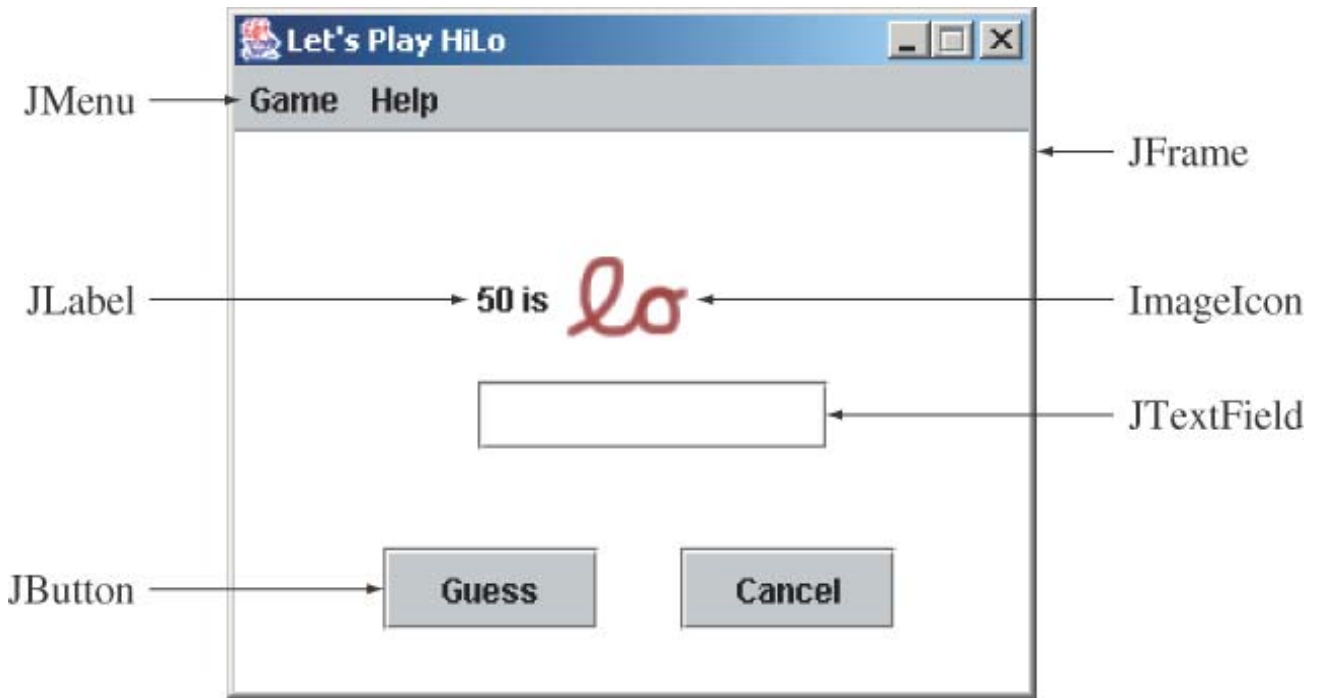
- Usar um modelo de programação por eventos (“Java’s delegation-based event model”)

Os componentes gráficos da linguagem Java estão definidos nos packages **java.awt** e **javax.swing** (surge com a versão **Java2 SDK1.2**)

As classes Swing permitem maior portabilidade entre diferentes sistemas operativos.

(totalmente implementadas em Java)

Alguns componentes gráficos do package javax.swing



Definição de uma janela (JFrame)

-Um objecto do tipo *JFrame* contém os elementos básicos para manipularmos uma janela: abrir, fechar, mover e redimensionar

- A uma janela poderemos adicionar outros componentes gráficos

```
import javax.swing.*;
```

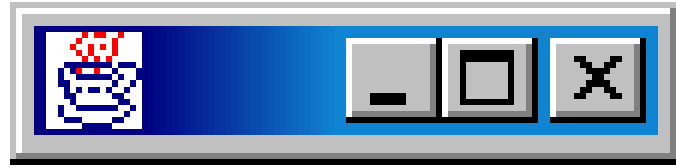
```
public class Janela {
```

```
public static void main (String [] args){
```

```
JFrame janela = new JFrame();
```

```
janela.setVisible(true);
```

```
}}
```



Alguns métodos da classe JFrame:

- atribuir um título

```
setTitle ("My First Subclass");
```

-dimensionar a janela

```
setSize ( 300 , 200 );
```

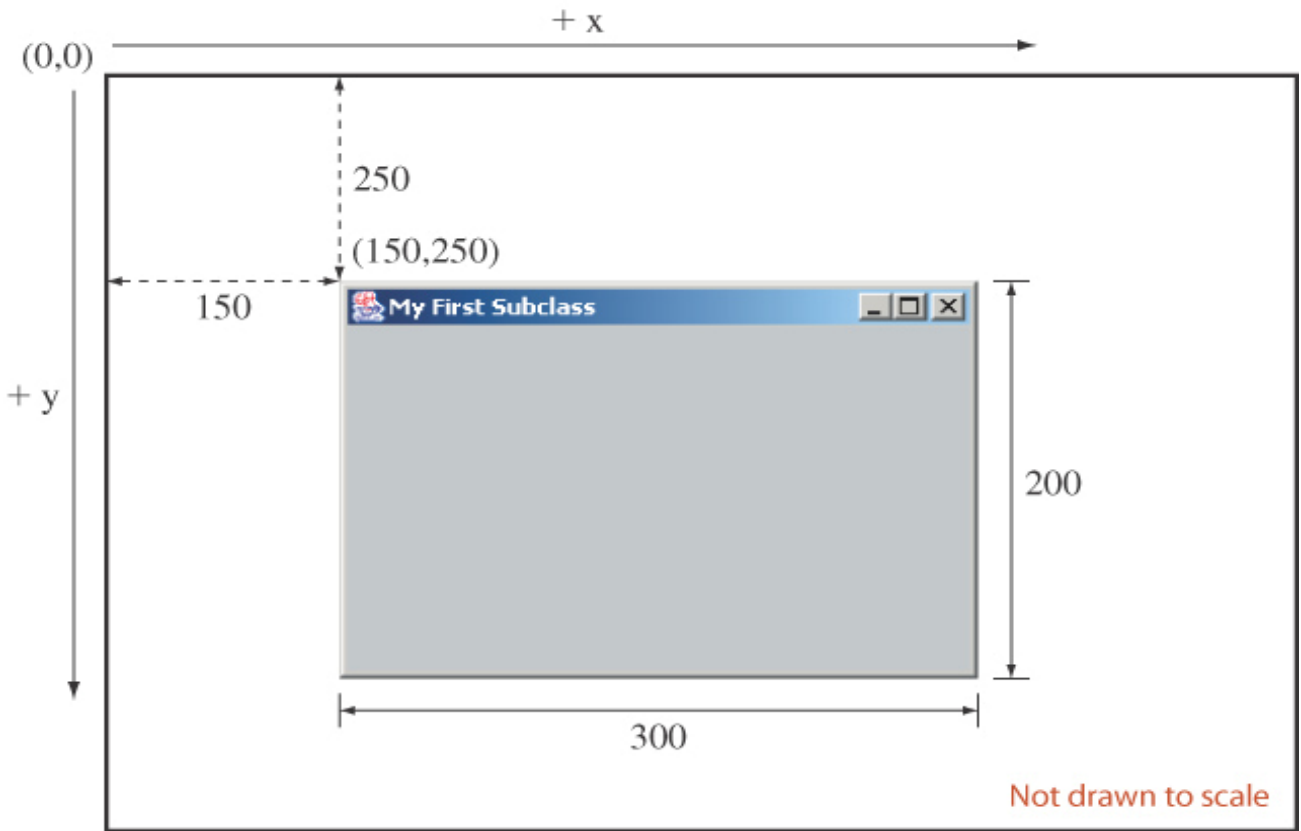
(300 pixels de largura e 200 pixels de altura)

- posicionar a janela no ponto de coordenadas (150, 250)

```
setLocation ( 150 , 250 );
```

- requerer que o programa termine quando a janela é fechada

```
setDefaultCloseOperation ( EXIT_ON_CLOSE );
```



Para construir uma interface gráfica podemos definir uma subclasse de JFrame. Nessa subclasse iremos adicionar o comportamento necessário à interface que pretendemos.

A classe Janela1 define uma janela com as características anteriores e com o fundo branco:

```
import javax.swing.*;

import java.awt.*;

public class Janela1 extends JFrame {

private static final int FRAME_WIDTH   = 300;
private static final int FRAME_HEIGHT  = 200;
private static final int FRAME_X_ORIGIN = 150;
private static final int FRAME_Y_ORIGIN = 250;

public Janela1(){
setTitle ( "Subclasse de JFrame com fundo branco" );
setSize (FRAME_WIDTH, FRAME_HEIGHT );
setLocation (FRAME_X_ORIGIN, FRAME_Y_ORIGIN );
setDefaultCloseOperation( EXIT_ON_CLOSE );
alterarCorFundo( );
}
```

```
private void alterarCorFundo() {
```

```
    Container contentor = getContentPane();
```

```
    contentor.setBackground(Color.white);
```

```
}
```

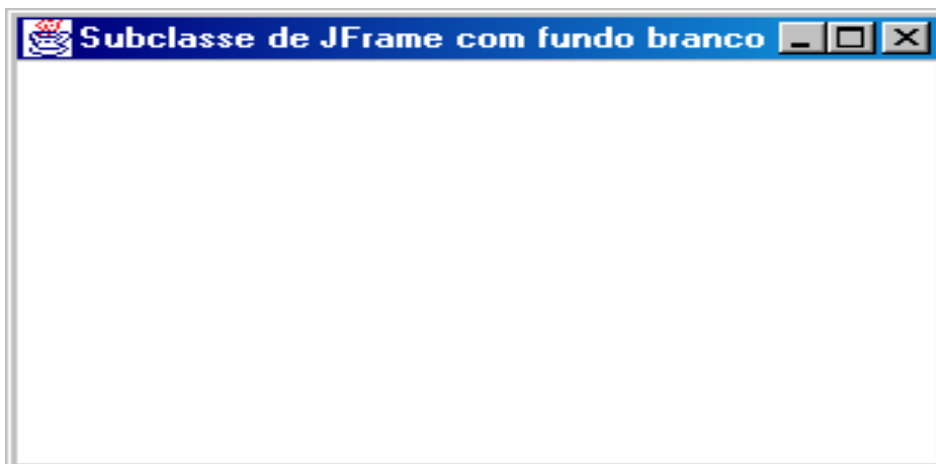
```
public static void main(String[] args) {
```

```
    Janela1 j = new Janela1();
```

```
    j.setVisible(true);
```

```
}
```

```
// fim da classe Janela1
```



Painel de conteúdos “*content pane*”

Designa a área da janela em que se pode mostrar conteúdo como texto, imagem, etc. (i. é, toda a área, com excepção do título, da barra de menus e do bordo da janela)

Para aceder ao painel de conteúdos de uma janela usamos o método:

Container getContentPane(),

definido na classe JFrame

O método devolve um objecto do tipo Container, classe do package java.awt

Vamos agora adicionar botões, objectos do tipo JButton, ao painel de conteúdos da nossa janela:

Há duas formas de colocar objectos gráficos no painel de conteúdos de uma janela:

1 - usar um objecto do tipo **LayoutManager** para controlar a colocação desses objectos

2 - explicitar o tamanho e a posição específica onde queremos colocar o objecto (posicionamento absoluto)

Para usarmos a segunda possibilidade devemos desactivar o layout manager, da classe Container, invocando o o método

void setLayout(LayoutManager mgr)

com o valor null

contentor.setLayout(null);

Para colocar o botão numa dada posição usamos o método da classe JButton:

void setBounds (x, y, width , height);

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

```
Container contentor = getContentPane();
```

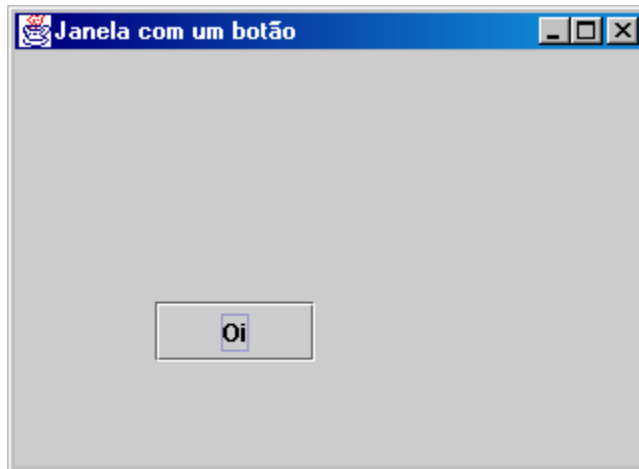
```
JButton botao = new JButton("Oi");
```

```
contentor.setLayout(null);
```

```
botao.setBounds ( 70, 125, 80, 30 );
```

```
//Finalmente, vamos colocar o botão na janela:
```

```
contentor.add( botao );
```



*Vamos agora ver como fazer o programa reagir
a um “click” sobre o botão ...*