

# TI-220 Java Orientado a Objetos

---

ANTONIO CARVALHO - TREINAMENTOS

A solid blue horizontal bar spanning the width of the slide, located at the bottom.

# Java FX

---

# Layouts

---

# Layouts

---

Os componentes gráficos podem ser dispostos no formulário exatamente na posição desejada definindo suas coordenadas cartesianas em *pixels*, através do método ***relocate( double X, double Y)***. Todo elemento gráfico que herda da classe ***javafx.scene.Node*** possui este método.

Porém como a aplicação Java é portátil e pode rodar em qualquer plataforma, o ideal é utilizar outros meios que não posicionem os elementos diretamente.

Para dispor os elementos gráficos na posição desejada de forma indireta, será preciso usar painéis com layouts pré-definidos.

# Layouts

---

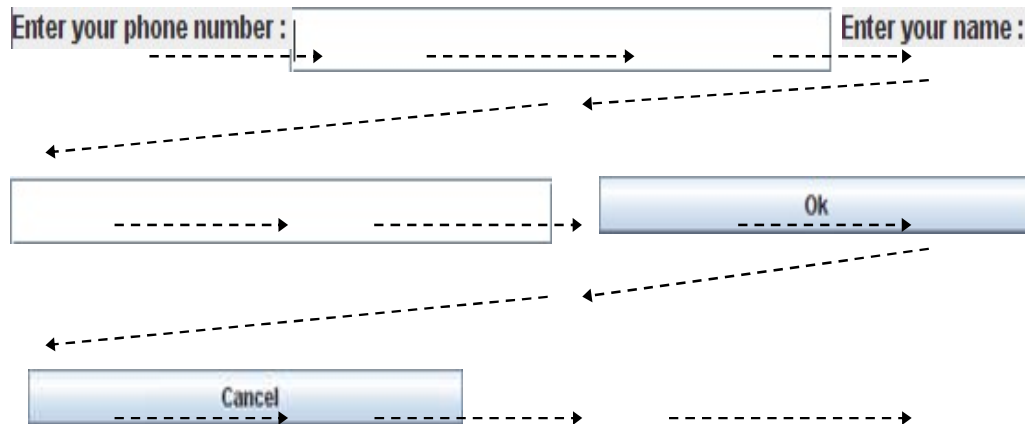
Os painéis com layouts pré-definidos posicionam seus **Nodes** filhos conforme o modelo do seu layout.

Existem diversos tipos de painéis com layout, e nos próximos slides serão abordados alguns tipos de *layouts* mais comuns (**BorderPane**, **HBox**, **VBox**, **StackPane**, **GridPane**, **FlowPane**, **TilePane**, **AnchorPane**)

# FlowPane

---

Este tipo de layout, posiciona os componentes filho um ao lado do outro, da esquerda para a direita de cima para baixo, respeitando o tamanho de cada componente filho.



- Caso não exista mais espaço na mesma linha o *FlowPane* cria mais uma linha.

# FlowPane

---

Para usar este tipo de painel, é preciso criar um **Pane** do tipo **FlowPane** e inserir os Nodes filhos na propriedade children.

- Sintaxe do uso do FlowPane:

- Criar o FlowPane

- Sintaxe:

- ```
FlowPane <objeto> = new FlowPane();
```

- Exemplo:

- ```
FlowPane panel = new FlowPane();
```

- Definir o Gap Horizontal com **setHgap()** ou Vertical **setVgap()**

- Sintaxe:

- ```
<objeto TilePane>.setHgap(int <distancia>);
```

- Exemplo:

- ```
panel.setHgap(4);
```

# FlowPane

---

E inserir os Nodes filhos na propriedade children.

- Adicionar os Nodes na propriedade children do FlowPane:

- Sintaxe:

- ```
<objeto FlowPane>.getChildren().add( <objeto Node> );
```

- Exemplo:

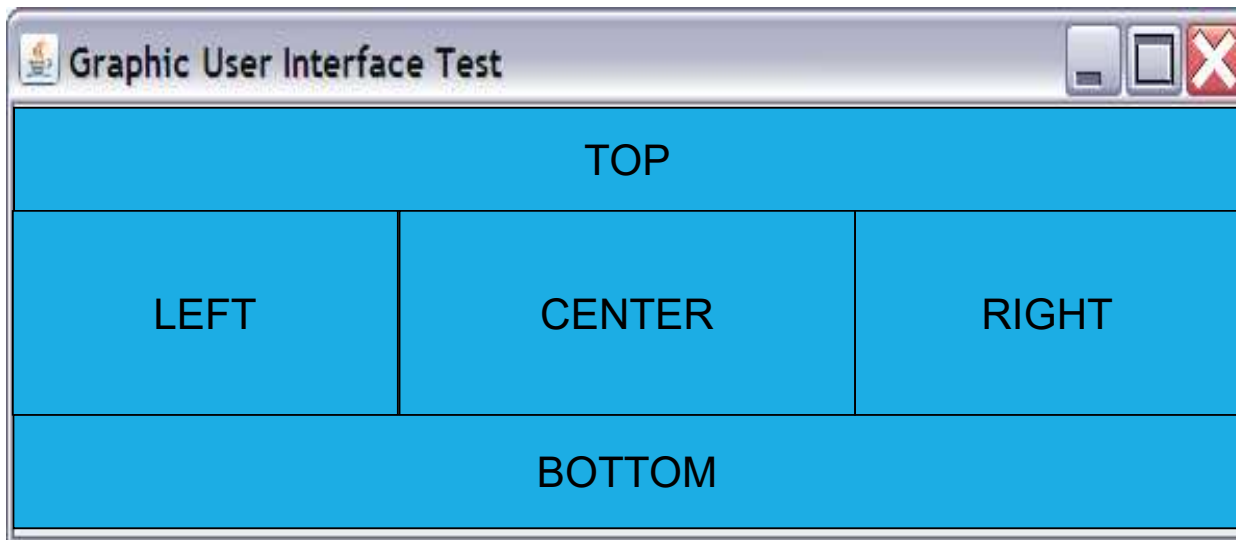
- ```
panel.getChildren().add( lblHello );
```



# BorderPane

---

BorderPane é um painel dividido em 5 zonas (Top, Bottom, Left, Right e Center), é possível colocar apenas um Node em cada uma destas zonas.



# BorderPane

---

Para usar este tipo de painel, é preciso criar um **Pane** do tipo **BorderPane** e inserir os Nodes filhos nas zonas desejadas com o uso dos métodos `setTop`, `setBottom`, `setLeft`, `setRight`.

- Sintaxe do uso do BorderPane:

- Criar o BorderPane

- Sintaxe:

- ```
BorderPane <objeto> = new BorderPane();
```

- Exemplo:

- ```
BorderPane pane2 = new BorderPane();
```

- Adicionar o Node na zona desejada do BorderPane:

- Sintaxe:

- ```
<objeto BorderPane>.setRight( <objeto Node> );
```

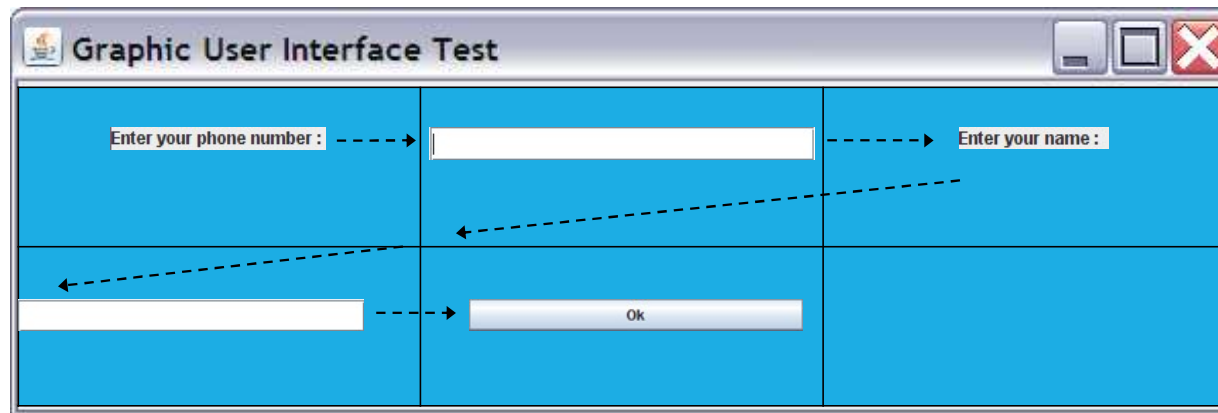
- Exemplo:

- ```
pane2.setRight( lblHello );
```

# TilePane

---

TilePane é um tipo de Painel que posiciona os Nodes da esquerda para a direita e de cima para baixo conforme o FlowPane, porém os Nodes ficarão todos do mesmo tamanho, pois será criada uma célula para cada Node.



# TilePane

---

Para usar o TilePane é preciso criar um objeto do tipo TilePane, definir a preferência de número de linhas ou colunas, e o espaçamento entre elas.

- Sintaxe do uso do TilePane:

- Criar o TilePane

- Sintaxe:

- ```
TilePane <objeto> = new TilePane();
```

- Exemplo:

- ```
TilePane pane3 = new TilePane();
```

- Definir a quantidade de colunas ou linhas:

- Sintaxe:

- ```
<objeto TilePane>.prefColumns( int colunas );
```

- Exemplo:

- ```
pane3.prefColumns( 3 );
```

# TilePane

---

Depois os componentes podem ser adicionados na propriedade children

- Definir o Gap Horizontal com **setHgap()** ou Vertical **setVgap()**

- Sintaxe:

- ```
<objeto TilePane>.setHgap(int <distancia>);
```

- Exemplo:

- ```
pane3.setHgap(4);
```

- Adicionar Nodes na propriedade **children**

- Sintaxe:

- ```
<objeto TilePane>.getChildren().add( <objeto Node> );
```

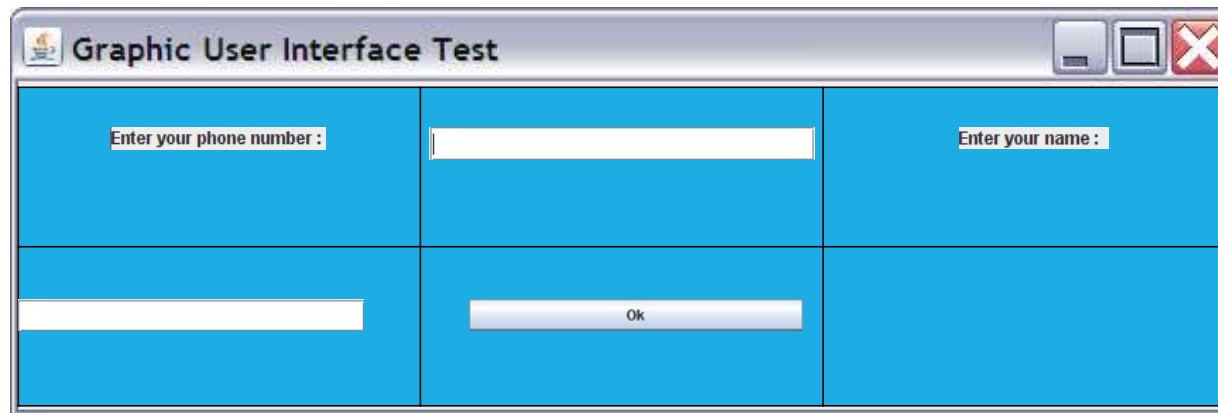
- Exemplo:

- ```
pane3.getChildren().add( lblHello );
```

# GridPane

---

GridPane é um tipo de Painel que posiciona os Nodes em uma grade, podendo escolher em qual célula o nó será colocado, e qual será seu Col Span e Row Span.



# GridPane

---

Para usar o GridPane é preciso criar um objeto do tipo GridPane, definir a quantidade de linhas e colunas, e o espaçamento entre elas (opcional).

- Sintaxe do uso do GridPane:

- Criar o GridPane

- Sintaxe:

- ```
GridPane <objeto> = new GridPane();
```

- Exemplo:

- ```
GridPane pane4 = new GridPane();
```

- Definir o Gap Horizontal com **setHgap()** ou Vertical **setVgap()**

- Sintaxe:

- ```
<objeto GridPane>.setHgap(int <distancia>);
```

- Exemplo:

- ```
pane4.setHgap(4);
```

# GridPane

---

Depois os componentes podem ser adicionados na propriedade children

- Adicionar Nodes na célula especificada

- Sintaxe:

- ```
<objeto GridPane>.add( <objeto Node>, int column, int row );
```

- Exemplo:

- ```
pane4.add( lblHello, 1, 2 );
```



# Exercício

---

Crie uma janela para cadastrar contatos, conforme o layout abaixo

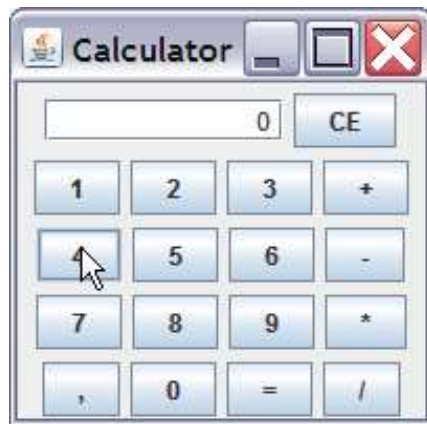


The image shows a window titled "Untitled" with standard window controls (minimize, maximize, close) in the top right corner. The window contains a form with three text input fields. The labels "Id", "Nome", and "Telefone" are positioned to the left of their respective input fields. At the bottom left of the window, there are two buttons: "Salvar" and "Pesquisar".

# Exercício

---

Faça uma tela gráfica que desenhe uma calculadora conforme uma figura abaixo



```
TilePane <objeto TilePane> = new TilePane();  
<objeto TilePane>.getChildren().add( <Node> );
```

```
BorderPane <objeto BorderPane> = new BorderPane();  
<obj BorderPane>.setTop( <Node> );  
<obj BorderPane>.setCenter( <Node> );
```