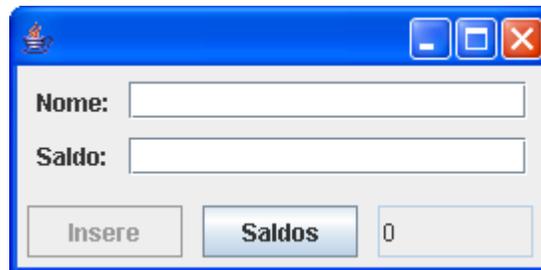


# Camada de Apresentação

- Programação Orientada ao Evento
  - Controle da aplicação pode estar na camada de interface
  - Aplicação limita-se a responder a eventos:
    - clicar do rato num botão
    - inserir um carácter num campo de texto
    - ...
  - Durante a inicialização da aplicação são registados os métodos que serão chamados quando ocorrerem determinados eventos.

## Camada de Apresentação

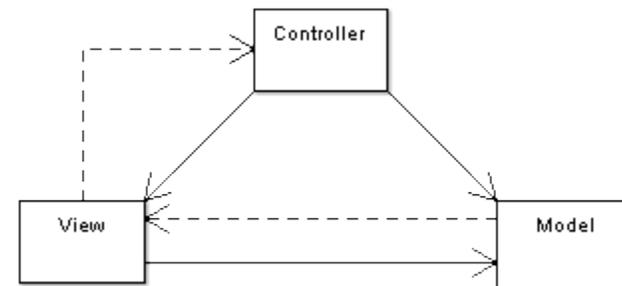
- Programação Orientada ao Evento
  - Que eventos podemos/temos de tratar neste exemplo?



A screenshot of a Windows-style dialog box with a blue title bar and standard window controls (minimize, maximize, close). The dialog contains two text input fields: the first is labeled "Nome:" and the second is labeled "Saldo:". Below the fields are three buttons: "Inserere", "Saldos", and a button with the number "0".

## Camada de Apresentação

- Model - View - Controller *pattern*



- Model: Business Logic
- View: A apresentação dos dados da aplicação. Desenho da interface e seus componentes
- Controller: Processa e responde aos eventos

## Camada de Apresentação

- Observer *pattern*
  - The **observer pattern** (sometimes known as [publish/subscribe](#)) is a [design pattern](#) used in computer programming to observe the state of an [object](#) in a [program](#). ([www.wikipedia.org](http://www.wikipedia.org)).
  - Através deste padrão podemos fazer com que a camada Business Logic gere eventos para a camada de interface, ou gerar interações entre componentes.

## Camada de Apresentação

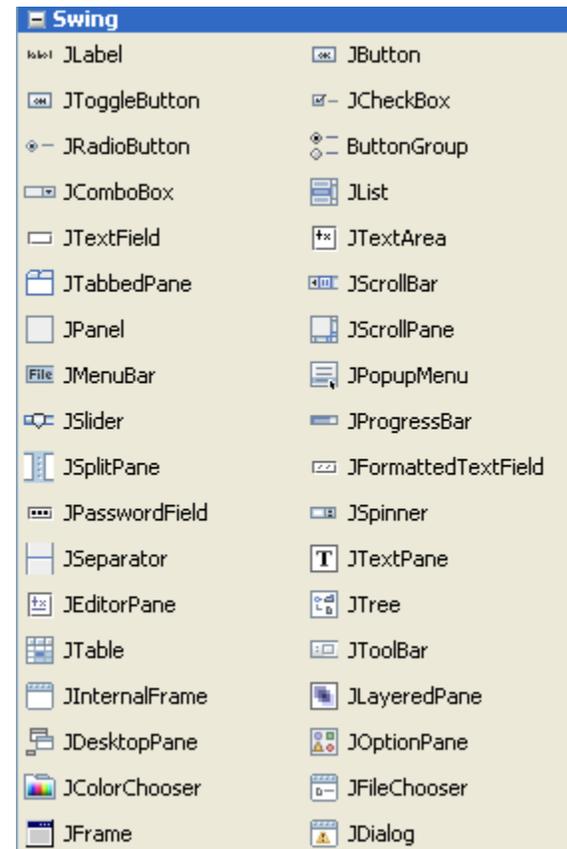
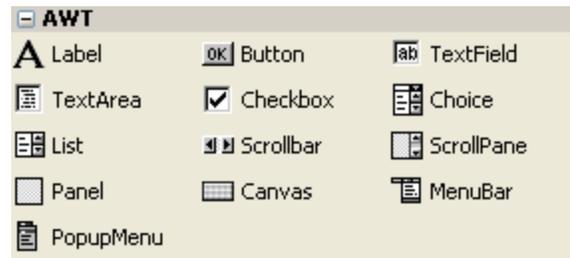
- Objecto *Observable* (class `java.util.Observable`)
  - Regista os seus observadores (Objectos *Observer*)  
`x.addObserver(y); // sendo x Observable e y Observer`
  - Define quando se encontra alterado  
`x.setChanged();`
  - Pede que os seus Observers sejam notificados  
`x.notifyObservers(Object arg);`
- Objecto *Observer* (interface `Observer`)
  - Recebe notificações de actualização: deve implementar o método  
`void update(Observable o, Object arg);`

# Camada de Apresentação

- Java GUI (Graphical User Interface)
  - Ligação entre a aplicação e o GUI nativo do sistema operativo
  - AWT: Abstract Window Toolkit
    - Código C nativo
    - Específico para cada plataforma
    - Limitado em certos aspectos (sem icons, tooltips, ...)
  - Swing (funciona sobre o AWT)
    - Java
    - Mais poderoso, sem limitações provocadas pelas plataformas
    - Look and Feel, Acessibilidade

# Camada de Apresentação

- Lista de componentes disponibilizados pelas duas APIs



# Camada de Apresentação

- Disposição de Componentes na Interface - Layout Manager
  - A colocação de componentes é definida por um gestor de espaço (layout manager)
  - Alguns gestores disponíveis:
    - Box
    - Grid
    - Gridbag
  - A posição final dos componentes é ditada pelo gestor (embora o utilizador possa fazer alguns pedidos)
  - É necessário precaver situações como por exemplo o redimensionamento da janela

## Referências

- Creating a GUI with JFC/Swing (aka the Swing Tutorial)
  - <http://java.sun.com/docs/books/tutorial/uiswing/index.html>

# Componentes e Listeners

- Pedaco da matriz que associa componentes aos respectivos listeners.

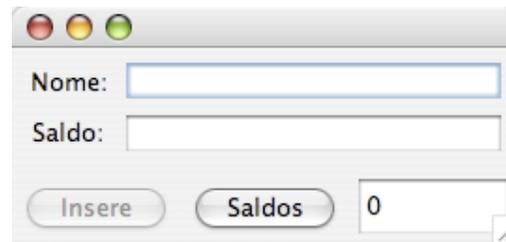
This table lists Swing components with their specialized listeners

Component	Listener							
	<a href="#">action</a>	<a href="#">caret</a>	<a href="#">change</a>	<a href="#">document, undoable edit</a>	<a href="#">item</a>	<a href="#">list selection</a>	<a href="#">window</a>	other
<a href="#">button</a>	✓		✓		✓			
<a href="#">check box</a>	✓		✓		✓			
<a href="#">color chooser</a>			✓					
<a href="#">combo box</a>	✓				✓			
<a href="#">dialog</a>							✓	
<a href="#">editor pane</a>		✓		✓				<a href="#">hyperlink</a>
<a href="#">file chooser</a>	✓							
<a href="#">formatted text field</a>	✓	✓		✓				
<a href="#">frame</a>							✓	
<a href="#">internal frame</a>								<a href="#">internal frame</a>
<a href="#">list</a>						✓		<a href="#">list data</a>
<a href="#">menu</a>								<a href="#">menu</a>
<a href="#">menu item</a>	✓		✓		✓			<a href="#">menu key</a> <a href="#">menu drag mouse</a>
<a href="#">option pane</a>								
<a href="#">password field</a>	✓	✓		✓				
<a href="#">popup menu</a>								<a href="#">popup menu</a>

## Exercício

- Para o exemplo que tem vindo a ser desenvolvido, substitua a classe de apresentação anteriormente feita, por uma em que se faça recurso a componentes Swing.

– Ex:



- Para saber mais sobre action listeners ver:
  - <http://java.sun.com/docs/books/tutorial/uiswing/events/actionlistener.html>
- Para saber como desenvolver aplicações com NetBeans ver:
  - <http://java.sun.com/docs/books/tutorial/uiswing/learn/settingup.html>