INTERFACE GRÁFICA DO JAVA

APTECH® COMPUTER EDUCATION Escola de Tecnologia

Objetivo do Curso

- Conhecendo a API (biblioteca) AWT
- Utilização da API (biblioteca) SWING
- Internacionalização
- Java Mail
- Sockets

O que é a biblioteca Swing?

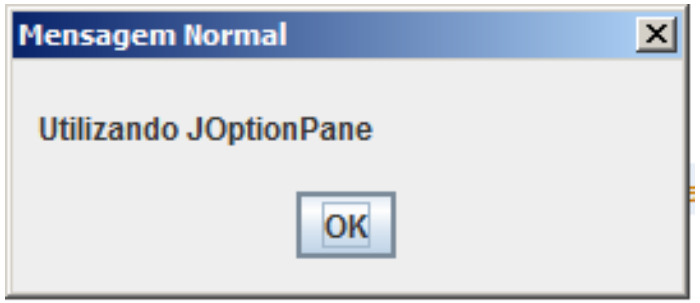
- **AWT** foi a primeira biblioteca (Conjunto de classes) gráfica criada para Java.
- **SWING** é a biblioteca gráfica mais utilizada para Java, por tem todas as características da AWT, é a mais atualizada.
- **GUI** : Graphical User Interface.

Possibilidades...

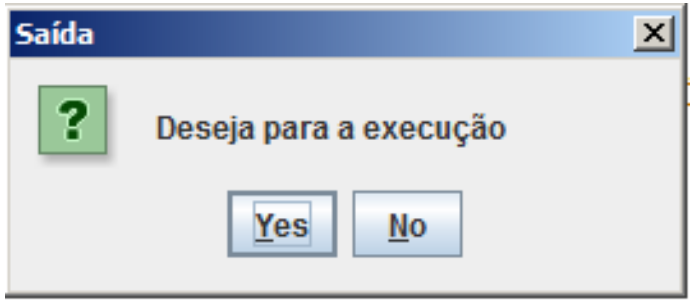
- Criar janelas, botões, campos de textos...
- Eventos, o que deve acontecer quando um usuário clicar em algo na tela.

Classe JOptionPane

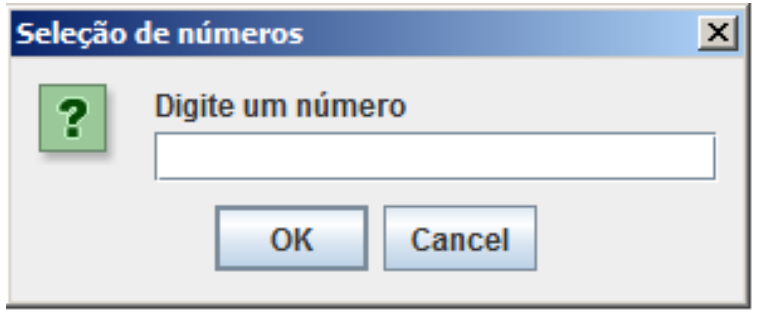
- Criação de caixas de diálogos simples e objetivas.



Caixa de diálogo
showMessageDialog()



Caixa de confirmação
showConfirmDialog()



Caixa de inserir texto
showInputDialog()

Exemplo 01 – Criando Caixas de Diálogos

- Criar 3 modelos de caixas de diálogos
  - showMessageDialog()
  - showConfirmDialog()
  - showInputDialog()
- Necessário

```
import javax.swing.JOptionPane;
```

Exemplo 01 – Criando Caixas de Dialogos

showMessageDialog()

Parâmetros

1. Null
2. Mensagem
3. Título
4. Tipo Mensagem

Exemplo 01 – Criando Caixas de Diálogos

showMessageDialog()

Tipos de Mensagem

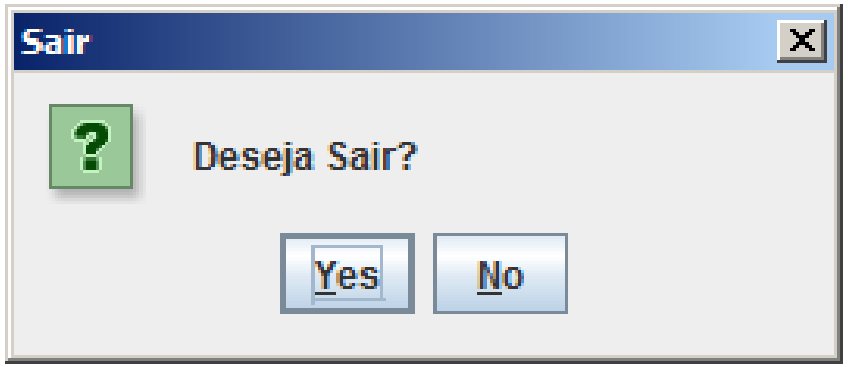
- JOptionPane.INFORMATION_MESSAGE
- JOptionPane.ERROR_MESSAGE
- JOptionPane.WARNING_MESSAGE
- JOptionPane.QUESTION_MESSAGE
- JOptionPane.PLAIN_MESSAGE

Exemplo 01 – Criando Caixas de Diálogos

showConfirmDialog()

Parâmetros

1. Null
2. Mensagem
3. Título
4. Botões
5. Tipo de Mensagem



Exemplo 01 – Criando Caixas de Diálogos

showConfirmDialog()

Botões

- JOptionPane.YES_NO_OPTION
- JOptionPane.YES_NO_CANCEL_OPTION
- JOptionPane.OK_CANCEL_OPTION
- JOptionPane.DEFAULT_OPTION

Exemplo 01 – Criando Caixas de Diálogos

showConfirmDialog()

Constantes

YES ou OK = 0

NO = 1

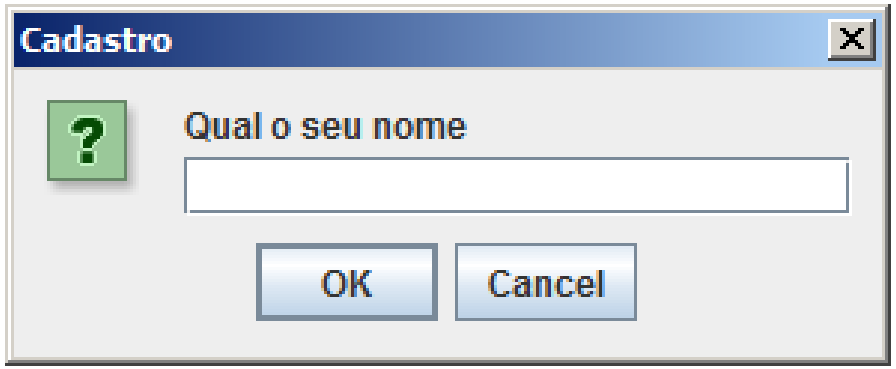
CANCEL = 2

Exemplo 01 – Criando Caixas de Diálogos

showInputDialog()

Parâmetros

1. Null
2. Mensagem
3. Título
4. Tipo de Mensagem



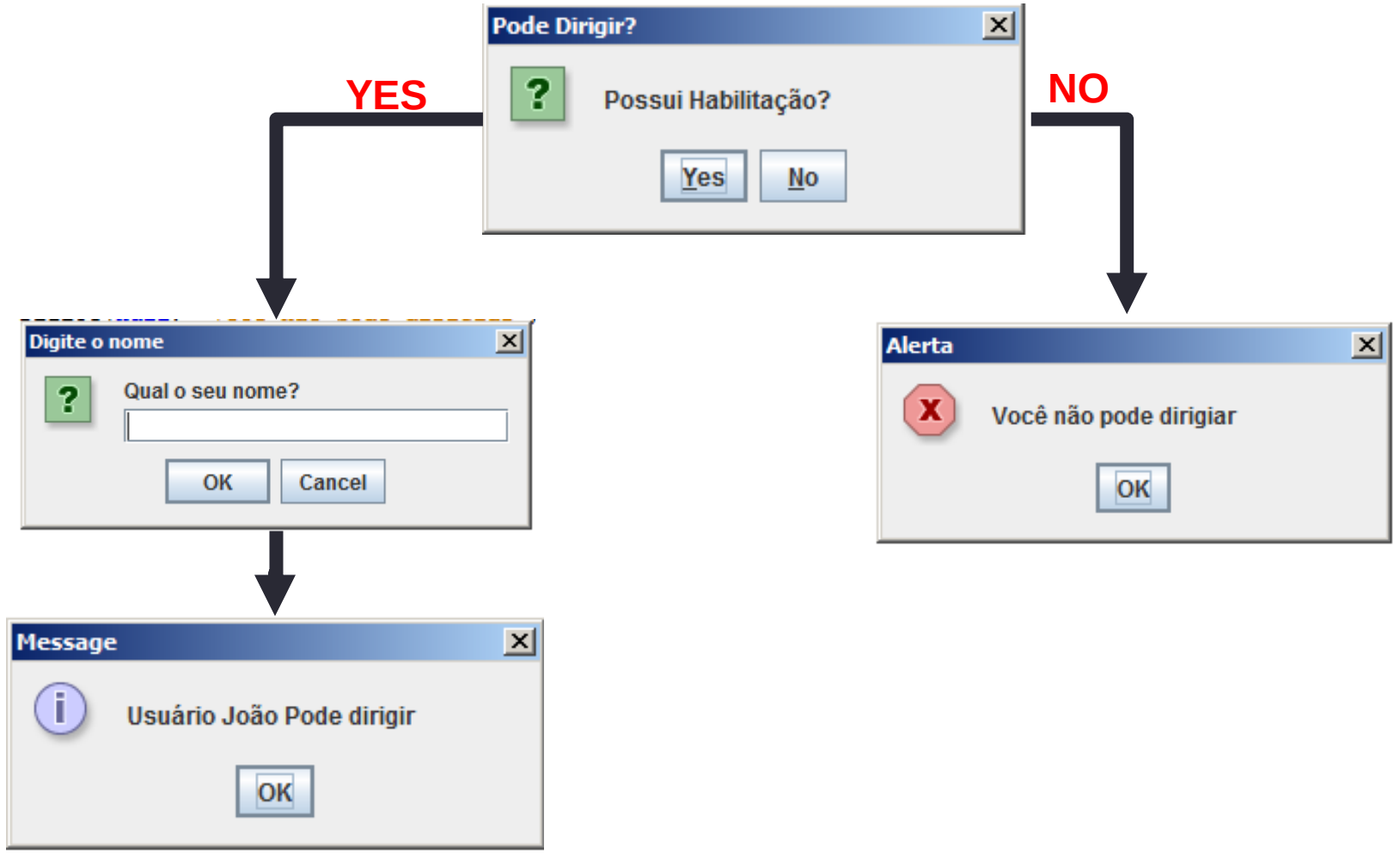
Conversão de tipos

```
//converter String para double
String valor = "123";
double valor2 = Double.valueOf(valor);

//converter String para int
String num = "5432";
int num2 = Integer.valueOf(num);

//converter qualquer coisa para String
int numero = 232222;
String numero2 = ""+numero;
```

Exercício 01

Crie uma GUI para ler um número inteiro e exibir o seu triplo.

Exercício 02



Java Swing

Os objetos gráficos no Java são criados como objetos normais a partir das classes definidas na biblioteca gráfica (neste caso Swing).

Para criar um elemento gráfico, basta instanciar um objeto do tipo escolhido.

Exemplo : Para criar uma caixa de texto é preciso instanciar um objeto da classe **JTextField** que é uma classe da biblioteca Swing e representa uma caixa de texto.

```
JTextField caixa = new JTextField();
```

Nota : Alguns tipos objetos gráficos podem receber parâmetros quando estiverem sendo instanciados.

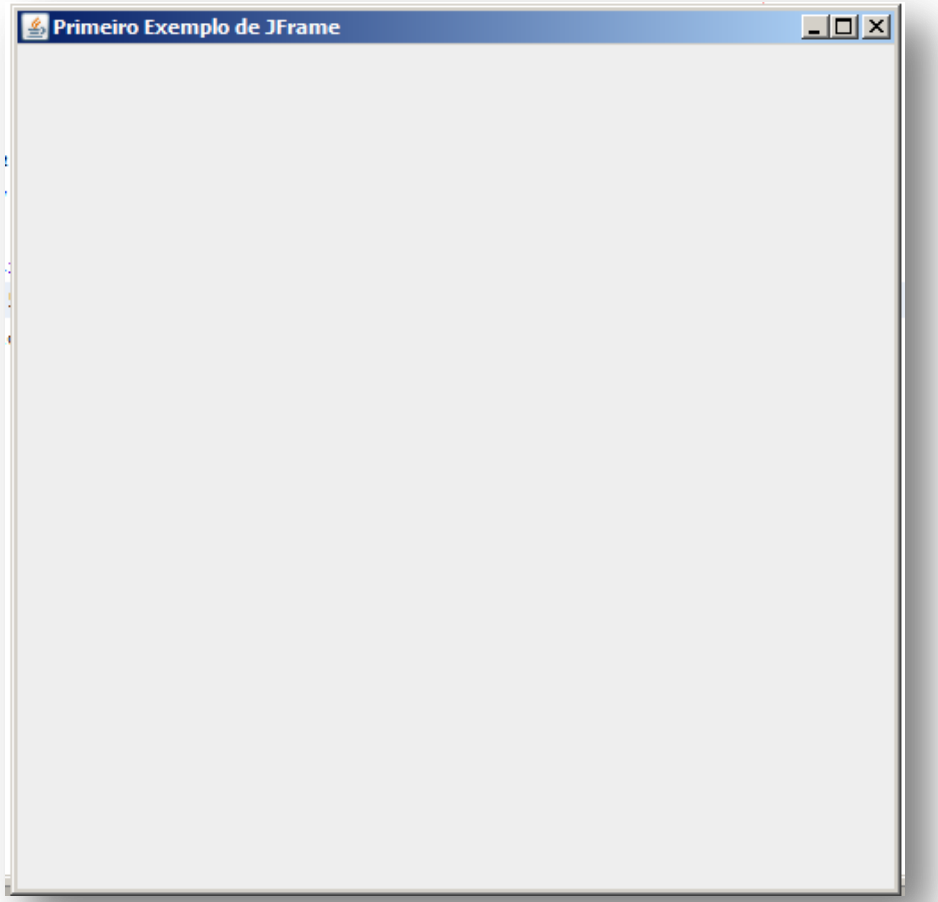
JFrame

- A classe JFrame é um recipiente para outros componentes do Swing.

- “Janela”

- **import javax.swing.JFrame;**

- **JFrame janela = new Jframe (“Nome da Janela”);**



Sintaxe

- Principais métodos
  - .setVisible() : Janela será visível ou não.
  - .setSize(L,A) : Tamanho da janela em pixels.
  - .setDefaultCloseOperation(): Ação padrão ao fechar a janela
  - .setLocationRelativeTo(): Posição do frame na tela
  - .setTitle(): Define o título da Janela
  - .setResizable(): Redimensionamento

*Obs.: Painel dentro da janela, componentes dentro do painel.

Exemplo 02 – Criando JFrame

Criar um JFrame com e utilizar os principais métodos.

1. Instanciar o Objeto.
2. Deixa-lo visível (true).
3. Definir tamanho (500, 500).
4. Ativar para fechar a aplicação ao clicar no [X].

(JFrame.EXIT_ON_CLOSE)

1. Opcional. Colocar nome na Janela (Título).

Exemplo:

```
JFrame janela = new JFrame();


janela.setTitle("Aula1 - JFrame"); //Título da Janela
janela.setVisible(true); //Visível igual a Verdadeiro
janela.setSize(800,600); //Definir o tamanho da Janela
janela.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE); //Ao clicar
no X, fechar o programa.
```

JPanel

- Como vimos, nosso frame está vazio. Só definimos o tamanho da 'moldura' e o título. Vamos adicionar alguns elementos ao nosso frame, para isso necessitamos colocar um painel, faremos isso inserindo um JPanel.

Sintaxe

- **import javax.swing.JPanel;**
- **JPanel painel = new JPanel();**
- Organização no padrão FlowLayout (esquerda para direita e de cima para baixo)

Exemplo 03 – Criando JPanel

Criar um JPanel

1. Instanciar o objeto JPanel.
2. Opcional Mudar a cor **painel.setBackground(color, cor)**
3. Adicionar componentes. **.add(componente)**
4. Adicionar painel na janela. **janela.add(painel);**

Exemplo 03 – Criando JPanel

```
JFrame janela = new JFrame();
JPanel painel = new JPanel();
janela.add(painel);
janela.setTitle("Aula1 - JFrame"); //Título da Janela
janela.setVisible(true); //Visível igual a Verdadeiro
janela.setSize(800,600); //Definir o tamanho da Janela
janela.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE); //Ao clicar
no X, fechar o programa.
```

JLabel

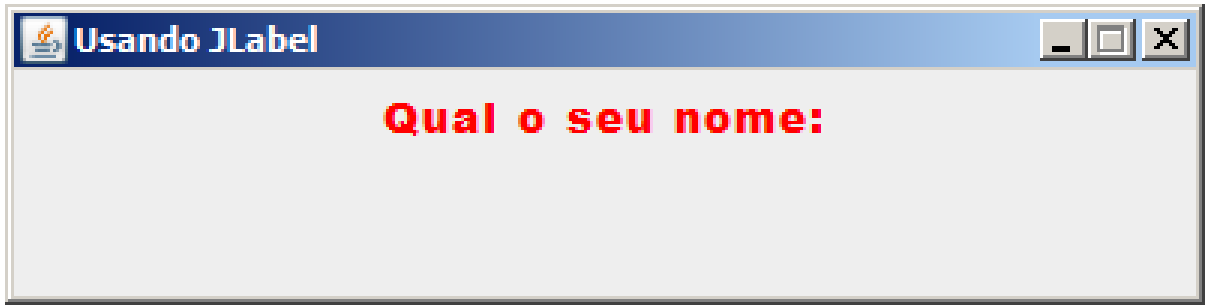
Um rótulo de texto. Um texto informativo que pode passar uma orientação ao usuário.

Existe 2 maneiras de criar:

1- JLabel texto = new JLabel();

texto.setText(“Texto”);

2- JLabel texto2 = new JLabel(“Texto”);



JButton - botões

Executa uma ação apenas quando pressionado.

Para utilizar o botão temos alguns passos.

1- Instanciar o objeto da classe JButton.

2- Adicionar uma ação do botão.

-botao.addActionListener(new ActionListener);

Dica: após o new digitando ac+ctrl+espaço, ecolher a opção **ActionListener**.

3- *Apagar o throw

4- Digitar o programa que será executado quando pressionar o botão.

Exemplo 04 - JButton

```
JButton b1 = new JButton("Não Aperte");
        b1.addActionListener(new ActionListener() {


            @Override
            public void actionPerformed(ActionEvent ae) {
                painel.setBackground(Color.yellow);
            }
        });
```

JButton - botões

Exercícios: 1- Criar um programa com 3 botões, cada um que pressionar o painel mudará a cor.

2- Criar um programa com 3 botões, deixando apenas o b1 visível, ao pressioná-lo, ele desaparece e o b2 aparece, ao pressionar o b2, ele desaparece e aparece o b3 e ao pressioná-lo, o ciclo volta ao b1.

3- Criar um programa com 1 botão, que ao ser pressionado, muda a cor da janela e ao pressiona-lo novamente, a cor anterior volta.

JTextField

O **JTextField** é um componente muito utilizado para a criação de formulários, no qual é necessário a inserção dos dados pelo teclado.

OBS: Todos os dados são em String, caso seja necessário, devemos efetuar a conversões de tipo.

JTextField

Para definir o tamanho utilizamos o método.

.setPreferedSize(new Dimension(120,30)); Primeiro parâmetro comprimento e o segundo a altura.

Para pegar o conteúdo digitado, utiliza-se o método **.getText();**

Para adicionar dado, utilizamos o método **.setText();**

Exemplo 05 – Criando JTextField

```
JTextField caixa1 = new JTextField();  //Criando caixa de texto

caixa1.setPreferredSize(new Dimension(130, 30));  //Definindo tamanho

String valor = caixa1.getText( ); //Pegando o Dados digitado.

caixa1.setText( ); //Atribuindo o dados na caixa.
```

Execícios JTextField

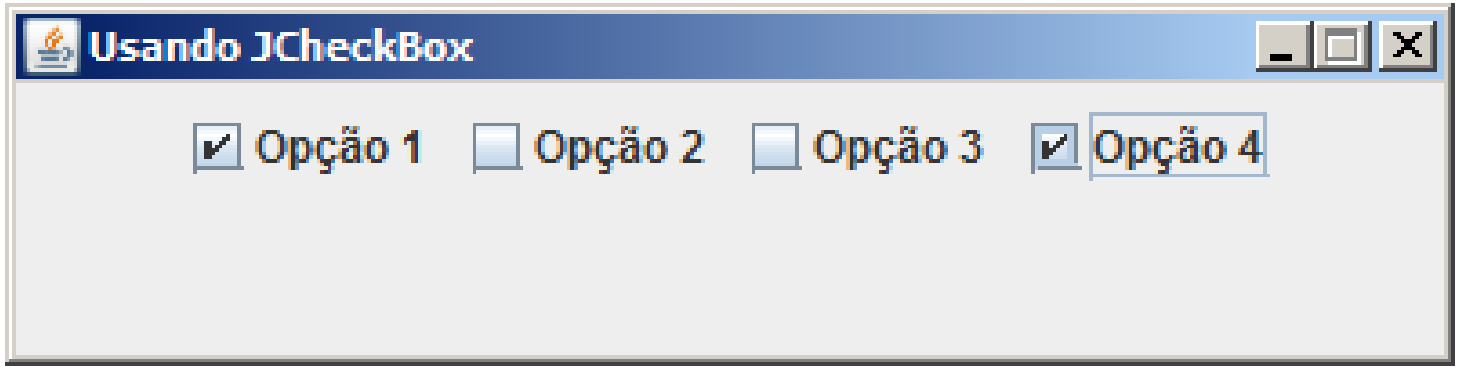
1- Criar um programa que pede a idade e verifica se a pessoa é maior ou menor de idade.

2- Criar um programa que contenha 2 caixas que solicite 2 valores, criar os botões “+”, “-”, “x” e “/ ”, conforme o botão pressionado, efetuar a operação correspondente.

JCheckBox

Caixa de seleção, permite selecionar mais de uma opção.

Para saber qual item está selecionado, utilizamos o método **.isSelected()** para cada item adicionado. Seu retorno é true ou false.



Exemplo 06 – Criando JCheckBox

```
JCheckBox box1 = new JCheckBox("op1");
JCheckBox box2 = new JCheckBox("op2");


if (box1.isSelected()){
    System.out.println("Opção 1");
}
if (box2.isSelected()){
    System.out.println("Opção 2");
}
```

Exercícios – JCheckBox

1- Criar um programa que contenha 3 JCheckBox e 1 botão para verificar. O termo só é valido caso os 3 termos estejam selecionados.

2- Criar um programa de pedido de um restaurante, o cliente terá as opções: **arroz**, **feijão**, **carne**, **salada** e **batata frita**.

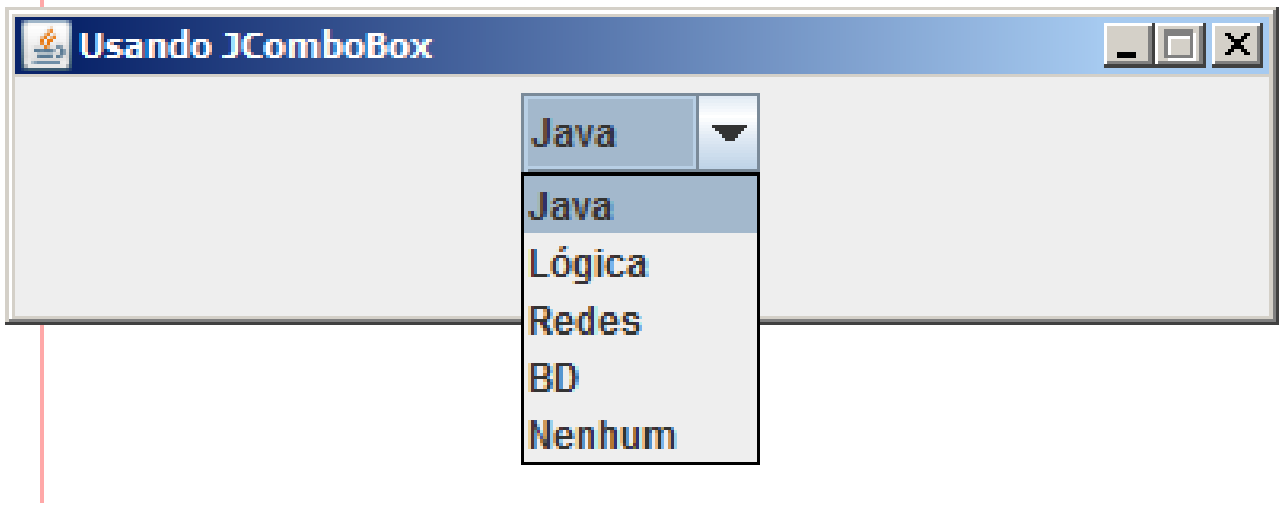
Após selecionar as opções, clicar no botão **pedido** para finalizar o mesmo. Mostrar a frase “**Pediu** + os itens selecionados”.

JComboBox

Uma caixa de combinação no qual o usuário pode selecionar UMA opção. Os dois métodos principais são:

.addItem( ); - Utilizado para adicionar um item na lista

.getSelectItem( ); - Utilizado para ler o item que foi selecionado.



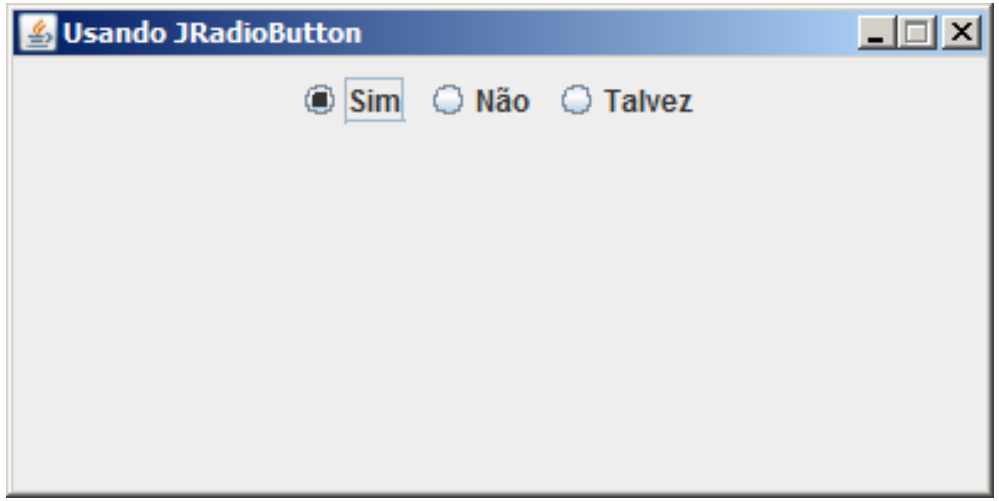
Exemplo 07 – Criando JComboBox

```
JComboBox combo1 = new JComboBox();

        combo1.addItem("Estados");

        combo1.addItem("AM");

        combo1.addItem("BA");


System.out.println("Item Escolhido: "+combo1.getSelectedItem());
```

JRadioButton

Permite a seleção exclusiva de um único valor dentre as opções definidas. Necessário criar um agrupador (**ButtonGroup**).

Assim como no JButton, o JRadioButton também permite uma ação ao ser selecionado um dos itens (***addActionListener...***).



Exemplo 08 – Criando JRadioButton

```
JRadioButton radio1 = new JRadioButton(“Solteiro");

JRadioButton radio2 = new JRadioButton("Casado");

JRadioButton radio3 = new JRadioButton("Divorciado");

ButtonGroup grupo1 = new ButtonGroup();

//adicionar radio no grupo

grupo1.add(radio1);

grupo1.add(radio2);

grupo1.add(radio3);
```

Exercício – JRadioButton

Criar um programa que tenha uma pergunta de múltipla escolha com 5 opções de resposta. Mostrar se a pessoa acertou ou errou.

JRadioButton

O que pode ser melhorado?

Quando se instala um programa no Windows percebemos que tem o termos e condições. Temos a opção Aceitar e Não Aceitar. Geralmente a opção aceitar já fica selecionada. Para que isso ocorra em nosso programa, utilizamos o método **.setSelect(true);**

JRadioButton

Exercício:

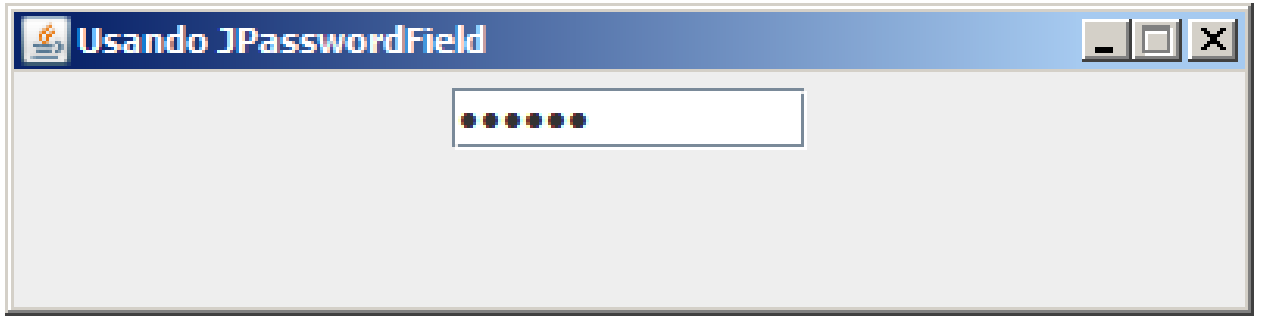
Criar um programa com a opção **Aceito** e **Não Aceito**, deixando a opção **Aceito** já selecionada, o botão **Continuar** só deve estar disponível quando a opção **Aceito** estiver selecionada, caso contrário, deverá estar indisponível.

JPasswordField

Campo de texto protegido é utilizado principalmente para senhas.

Não se utiliza o método **.getText()** para esse componente, pois se trata de um campo protegido.

Para utilizar o JPassowoedField temos que criar uma String e salvar os dados lido convertido em String.



Exemplo 09 – Criando JPasswordField

```
JPasswordField caixa1 = new JPasswordField();
caixa1.setPreferredSize(new Dimension (130,40));

JButton b1 = new JButton("Senha");

String senha;
senha = String.valueOf(caixa1.getPassword());
```

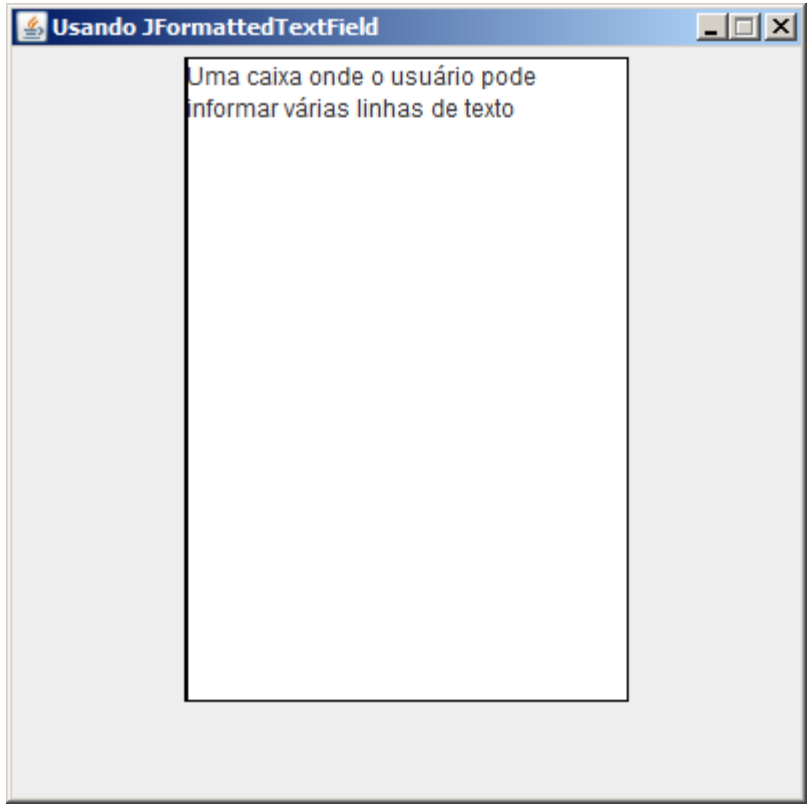
Exercício – Criando JPasswordField

Criar um programa no qual o usuário digita o login e a senha, o programa terá que dizer se o acesso está liberado ou negado após ser pressionado o botão **entrar**.

JTextArea

Uma caixa na qual o usuário pode informar várias linhas de texto.

Assim como o JTextField, utilizamos o método **.getText();** para ler os dados digitados.



Exemplo 10 – Criando JTextArea

```
JTextArea area = new JTextArea();

area.setPreferredSize(new Dimension(250, 70));

String valor = area.getText();
System.out.println("Valor  "+ valor);
```