



INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
RIO GRANDE DO NORTE

AULA:

Interfaces Gráficas em Java (+ Strings)

Programação Orientada a Objetos

Alba Lopes, Profa.

<http://docentes.ifrn.edu.br/albalopes>
alba.lopes@ifrn.edu.br

Pacotes da API Java

- ▶ A Java API (*Application Programming Interface* - ou Interface de Programação de Aplicações) oferece uma rica coleção de classes e métodos para:
 - ▶ realizar cálculos matemáticos comuns
 - ▶ manipular strings
 - ▶ verificação de erros
 - ▶ Construção de interfaces gráficas, etc
- ▶ Muitas classes já são predefinidas, agrupadas em diretórios no disco, que podemos nos referir como pacotes



Pacotes da API Java

- ▶ Para utilizar uma classe já existente na API Java, devemos utilizar uma instrução do tipo **import**.
- ▶ Por exemplo: para utilizar a classe **Scanner**, do pacote **java.util**, devemos fazer:

```
import java.util.Scanner;
```

- ▶ Um grande poder de Java é a grande quantidade de classes nos pacotes da Java API que os programadores podem reutilizar ao invés de construir do zero.
- ▶ Você pode consultar a lista dos pacotes da API Java visitando o endereço:
<http://docs.oracle.com/javase/6/docs/api/>



Java Swing

- ▶ A biblioteca Swing faz parte do pacote **javax.swing**
- ▶ Possui um Conjunto de componentes GUI (Graphical User Interface) que simplificam o processo de desenvolvimento de interfaces gráficas
 - ▶ botões, caixas de texto, painéis, rótulos e todos os outros componentes gráficos



Primeiro Exemplo

- ▶ Crie um novo pacote com o nome aula08
- ▶ Crie um arquivo OlaMundo.java que possua o seguinte código:

```
public class OlaMundo {  
  
    public static void main(String[] args) {  
        JOptionPane.showMessageDialog(null, "Olá Mundo!");  
    }  
}
```

Salve e aperte
SHIFT+F6 para
executar



Componente JOptionPane

- ▶ Utilizamos o método `showMessageDialog` da classe `JOptionPane` para exibir mensagens na tela
- ▶ Para entrada de dados, o método a ser utilizado é o `JOptionPane.showInputDialog`.
- ▶ Esse método solicita um parâmetro e retorna o valor digitado pelo usuário

```
public class ExemploEntrada{  
    public static void main(String [] args){  
        String nome = JOptionPane.showInputDialog("Digite o seu nome: ");  
        JOptionPane.showMessageDialog(null, "Olá " + nome + "!");  
    }  
}
```



Componente JOptionPane

- ▶ O valor recebido com o JOptionPane é sempre do tipo String! Porém, é possível realizar a conversão de tipos.
- ▶ Convertendo para inteiro

```
String texto = JOptionPane.showInputDialog("Digite um número inteiro:");  
int x = Integer.parseInt(texto);
```

- ▶ Convertendo para float

```
String texto = JOptionPane.showInputDialog("Digite um número float:");  
float y = Float.parseFloat(texto);
```

- ▶ Convertendo para double

```
String texto = JOptionPane.showInputDialog("Digite um número double:");  
double z = Double.parseDouble(texto);
```



Componente JOptionPane

▶ Exemplo Soma:

```
public class Soma{  
    public static void main(String [] args){  
        int n1, n2, resultado;  
        String entrada = JOptionPane.showInputDialog("Digite o número 1: ");  
        n1 = Integer.parseInt(entrada);  
        entrada = JOptionPane.showInputDialog("Digite o número 2: ");  
        n2 = Integer.parseInt(entrada);  
        resultado = n1 + n2;  
        JOptionPane.showMessageDialog(null, "O resultado da soma é: " + resultado);  
    }  
}
```



Exercícios

1. (AreaRetangulo.java) Calcular e mostrar o valor da área de um retângulo dado o valor dos dois lados. A área do retângulo é calculada pela fórmula
 - ▶ $\text{areaRetangulo} = \text{lado1} * \text{lado2}$
2. (Imc.java) Faça um programa que leia o peso e a altura de uma pessoa e mostre o índice de massa corporal (IMC). O IMC é calculado da seguinte forma:
 - ▶ $\text{imc} = \text{peso} / (\text{altura} * \text{altura})$



Classe String

- ▶ String é uma classe JAVA que faz parte do pacote `java.lang.String`
- ▶ São tratadas como se fossem tipos primitivos (como `int`, `float`, `boolean`)
- ▶ São criadas de duas formas

- ▶ Através de construtores

```
String s = new String("nova String");
```

- ▶ Através da atribuição de um literal

```
String s = "nova String";
```



Classe String

- ▶ O operador de concatenação (+) é utilizado para unir String

```
public class Concatenar {  
  
    public static void main(String[] args) {  
        String a = "Nova ";  
        String b = "Cruz";  
        String cidade = a + b;  
  
        System.out.println(cidade);  
    }  
}
```



Classe String

- ▶ String são comparadas através do método `.equals()`

```
public static void main(String[] args) {  
    String a = "teste";  
    String b = "teste";  
  
    if (a.equals(b)) {  
        System.out.println("String iguais!");  
    } else {  
        System.out.println("String diferentes!");  
    }  
  
}
```



Classe String

- ▶ Retornar o tamanho da string: `.length()`

```
String cidade = "Nova Cruz";
```

```
System.out.println("A string tem: " + cidade.length() + " caracteres.");
```

- ▶ Retorna o índice do caractere `c`: `indexOf(char c)`

- ▶ As strings começam a contar do caractere 0:

N	o	v	a		C	r	u	z
0	1	2	3	4	5	6	7	8



Classe String

```
13 public static void main(String[] args) {
14
15     String cidade = "Nova Cruz";
16
17     System.out.println("O caractere C está na posição: " + cidade.indexOf('C'));
18     System.out.println("O caractere M está na posição: " + cidade.indexOf('M'));
19
20 }
```

Saída - ExemplosNovaCruz (run) Tarefas

```
run:
O caractere C está na posição: 5
O caractere M está na posição: -1
CONSTRUÍDO COM SUCESSO (tempo total: 1 segundo)
```

► Retorna o caractere na posição indicada: `.charAt(int posicao)`

```
13 public static void main(String[] args) {
14
15     String cidade = "Nova Cruz";
16
17     System.out.println("O caractere na posição 3 é: " + cidade.charAt(3));
18
19 }
20 }
```

Saída - ExemplosNovaCruz (run) Tarefas

```
run:
O caractere na posição 3 é: a
CONSTRUÍDO COM SUCESSO (tempo total: 1 segundo)
```



Classe String

▶ Outros métodos

▶ toLowerCase()

- ▶ Retorna nova String toda minúscula

▶ toUpperCase()

- ▶ Retorna nova String toda maiúscula

▶ trim()

- ▶ Retorna nova String sem os espaços no início e fim

▶ compareTo()

- ▶ Compara duas strings. Retorna: 0 se as strings forem iguais; valor maior do que 0 se a string for maior; valor menor que 0 a string for menor.

▶ compareToIgnoreCase(String str)

- ▶ Compara duas strings ignorando maiúsculas e minúsculas.



Classe String

▶ Outros métodos

▶ `replace(char caractere_antigo, char novo_caractere)`

- ▶ Retorna uma nova string substituindo todas as ocorrências do `caractere_antigo` pelo `caractere_novo`

▶ `substring(int inicio, int fim)`

- ▶ Retorna uma nova string que é parte da string original, delimitada pelos índices passados como parâmetro.

▶ Consulte a API para ver os outros métodos

- ▶ <http://docs.oracle.com/javase/7/docs/api/java/lang/String.html>



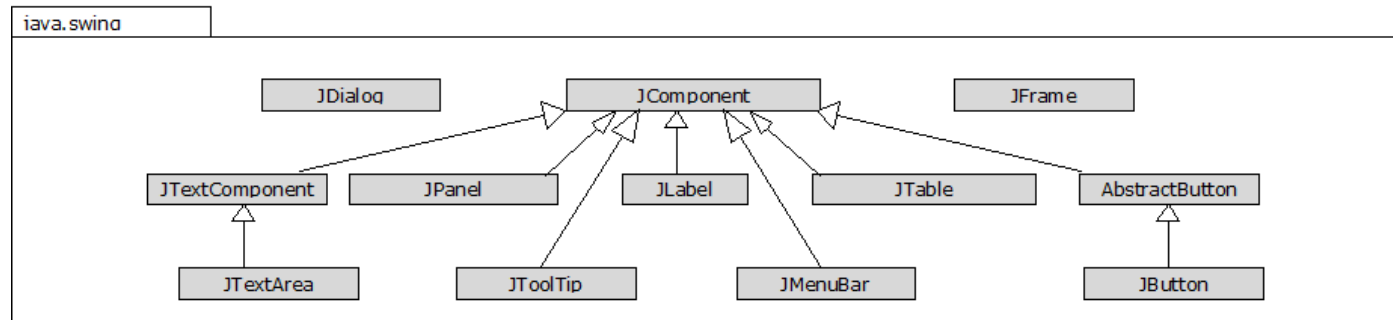
Exercícios

3. (Maiusculo.java) Receba o nome do usuário através de uma caixa de diálogo e apresente o valor lido todo em letras maiúsculas.
4. (Substituir.java) Leia uma frase do usuário e substitua onde o usuário tiver digitado a letra “a” por um @.

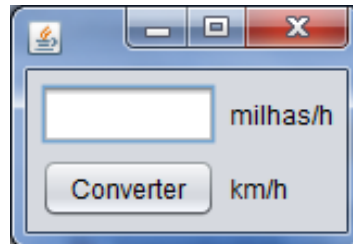


Swing

- ▶ Há muitos componentes disponíveis na biblioteca Swing para geração de interface gráfica.



- ▶ É possível gerar aplicações simples e elaboradas



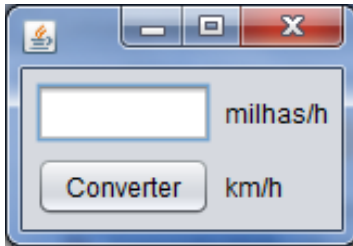
Swing - Componente JFrame

- ▶ O Componente JFrame é o componente de formulário.
- ▶ Os formulários em Java que iremos construir, herdarão da classe JFrame.
- ▶ Esse componente é que irá armazenar todos os componentes visuais necessários em nossa aplicação.



Swing - Componente JFrame

- ▶ O código fonte referente a essa aplicação será algo como:



```
public class ExemploJanela extends JFrame{
    private Container c;
    private JTextField t1;
    private JLabel l1, l2;
    private JButton b1;

    public ExemploJanela(String titulo){

        Container c = getContentPane();
        b1 = new JButton("Converter");
        l1 = new JLabel("milhas/h");
        l2 = new JLabel("km/h");
        t1 = new JTextField();
        t1.setColumns(7);

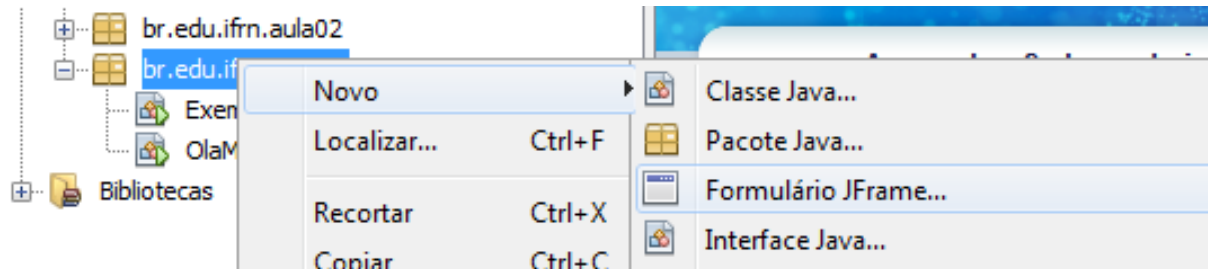
        c.setLayout(new FlowLayout());
        c.add(b1);
        c.add(t1);
        c.add(l1);
        c.add(b1);
        c.add(l2);
        setTitle(titulo);
        setSize(200, 100);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setVisible(true);
    }

    public static void main(String [] args){
        ExemploJanela c = new ExemploJanela("Converter Velocidade");
    }
}
```



Swing - Componente JFrame

- ▶ Felizmente, o NetBeans nos oferece mecanismos de criarmos facilmente aplicações que utilizam interface gráfica
- ▶ Siga o passo a passo do Tutorial 01 no site da disciplina para criar seu primeiro projeto usando formulário JFrame.



Referências

- ▶ DEITEL. Java - Como Programar. 6ª Edição. Editora Pearson.
- ▶ Na internet
 - ▶ <http://java.sun.com/products/jfc/reference/faq/index.html>
 - ▶ Metr pole Digital - http://www.metroledigital.ufrn.br/aulas_avancado/web/disciplinas/desktop
 - ▶ Netbeans - Introdu  o   Constru  o de GUIs - http://netbeans.org/kb/docs/java/gui-functionality_pt_BR.html

