

Programação Orientada a Objetos

ITE - 003

profº Mauricio Conceição Mario

//Funções Matemáticas

```
class Matematica {  
    public static void main (String args[]) {  
        double A=5.09, B=6.01;  
  
        /*função Math.ceil(): realiza o arredondamento de um número;  
        variável retornada também é do tipo double;  
        usada somente para double; aproxima para o próximo inteiro*/  
        System.out.print("5.09 arredondado para mais - ceil=" + Math.ceil(A) );  
  
        /*função Math.floor(): aproxima para o inteiro anterior*/  
        System.out.print("5.09 arredondado para menos - floor=" + Math.floor(A) );  
  
        /*função Math.max(): verifica o maior valor entre dois números do tipo double, int ou long */  
        System.out.print("o maior numero entre 5.09 e 6.01 - max=" + Math.max(A,B) );  
  
        /*função Math.min(): verifica o menor valor entre dois números do tipo double, int ou long */  
        System.out.print("o menor numero entre 5.09 e 6.01 - min=" + Math.min(A,B) );  
  
        /*função Math.sqrt(): calcula a raiz de um número do tipo double */  
        System.out.print("a raiz de 6.01 - sqrt=" + Math.sqrt(B) );  
  
        double C=2.001; int D=-3;  
  
        /*função Math.pow(): calcula a potência de um número elevado a outro */  
        System.out.print("2.001 elevado a -3 - pow=" + Math.pow(C,D) );  
  
        /*função Math.random(): retorna um número aleatório dentro de um valor-tipo double */  
        System.out.print("randomico entre 0 e 100 - random=" + (int) (Math.random() * 100) );  
    }  
}
```

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
```

```
public class Calculadora2 extends JFrame implements ActionListener {
```

```
    JLabel L1,L2,L3;
    JButton B1, B2, B3, B4, B5, B6, B7, B8, B9;
    JTextField T1, T2, T3;
```

Declaração de objetos
do tipo swing

```
    public static void main (String args[]) {
        JFrame janela2 = new Calculadora2();
        janela2.show();
        WindowListener x = new WindowAdapter() {
            public void windowClosing (WindowEvent fecha)
                { System.exit(0); }
        };
        janela2.addWindowListener(x);
    }
```

janela2, objeto do tipo JFrame, é
instanciado ao método construtor.
Através do objeto **janela2**,
é invocado o método para mostrar
a interface com o usuário.
Objeto **x**, registrado junto à
classe “ouvidora”, possibilita o uso
método que receba como
parâmetro eventos do *mouse*.

Calculadora2()

Método construtor: configuração da interface com usuário

```
{
    setTitle("AULA DE PROGRAMAÇÃO ORIENTADA A OBJETOS");
    setSize(400,120);
    getContentPane().setBackground (new Color (0,0,0));
    getContentPane().setLayout (new GridLayout(4,4));
    L1 = new JLabel("numero 1");
    L1.setForeground(Color.green);
    L1.setFont(new Font(" ",Font.BOLD,14));
    L2 = new JLabel("numero 2");
    L2.setForeground(Color.blue);
    L2.setFont(new Font(" ",Font.BOLD,14));
    L3 = new JLabel("resultado");
    L3.setForeground(Color.yellow);
    L3.setFont(new Font(" ",Font.BOLD,14));
    B1 = new JButton("+");
    B1.setBackground(Color.black);
    B1.setForeground(Color.white);
    B2 = new JButton("-");
    B2.setBackground(Color.black);
    B2.setForeground(Color.white);
    B3 = new JButton("*");
```

Definição de dimensões, título, e perfil de construção da interface

Instância e configuração dos componentes
Button e JLabel: texto, cor de fundo, fonte

```
B3.setBackground(Color.black);  
B3.setForeground(Color.white);  
B4 = new JButton("/");  
B4.setBackground(Color.black);  
B4.setForeground(Color.white);  
B5 = new JButton("limpar");  
B5.setBackground(Color.blue);  
B5.setForeground(Color.white);  
B6 = new JButton("potência");  
B6.setBackground(Color.blue);  
B6.setForeground(Color.orange);  
B7 = new JButton("raiz");  
B7.setBackground(Color.blue);  
B7.setForeground(Color.green);  
B8 = new JButton("inverso");  
B8.setBackground(Color.blue);  
B8.setForeground(Color.pink);  
B9 = new JButton("aleatório");  
B9.setBackground(Color.yellow);  
B9.setForeground(Color.blue);
```

Configuração dos componentes
Button e JLabel: texto, cor de
fundo, fonte

```
B1.addActionListener(this);  
B2.addActionListener(this);  
B3.addActionListener(this);  
B4.addActionListener(this);  
B5.addActionListener(this);  
B6.addActionListener(this);  
B7.addActionListener(this);  
B8.addActionListener(this);  
B9.addActionListener(this);
```

Adição dos objetos JButton ao método que “ouve” eventos do *mouse*

```
T1 = new JTextField();  
T1.setBackground(Color.black);  
T1.setForeground(Color.white);  
T2 = new JTextField();  
T2.setBackground(Color.black);  
T2.setForeground(Color.white);  
T3 = new JTextField();  
T3.setBackground(Color.black);  
T3.setForeground(Color.white);  
T3.setEditable(false);
```

Instância e configuração das áreas de texto

```
getContentPane().add(L1);  
getContentPane().add(T1);  
getContentPane().add(B1);  
getContentPane().add(B2);  
getContentPane().add(L2);  
getContentPane().add(T2);  
getContentPane().add(B3);  
getContentPane().add(B4);
```

Ordem em que os objetos correspondentes aos componentes são inseridos à interface JFrame, de acordo com a opção
`getContentPane().setLayout (new GridLayout(4,4));`

```
getContentPane().add(L3);
getContentPane().add(T3);
getContentPane().add(B5);
getContentPane().add(B6);
getContentPane().add(B7);
getContentPane().add(B8);
getContentPane().add(B9);
} // fim do método construtor
```

Ordem em que os objetos correspondentes aos componentes são inseridos à interface JFrame, de acordo com a opção
`getContentPane().setLayout (new GridLayout(4,4));`

```
public void actionPerformed(ActionEvent e) {
```

```
    if (e.getSource()==B5)
    {
        T1.setText(" ");
        T2.setText(" ");
        T3.setText(" ");
        return;
    }
```

Método actionPerformed() recebe como parâmetro o evento do *mouse*.
Condição do JButton B5 ser “clicado” corresponde à ação de limpar as áreas de texto correspondentes.

```
double n1=0, n2=0, resultado=0;
try
{   n1 = Double.parseDouble(T1.getText());
  n2 = Double.parseDouble(T2.getText()); }
catch (NumberFormatException erro)
{ T3.setText("erro"); return; }
```

O que for digitado nas áreas de texto T1 e T2 será transformado em número real; o **try...catch{ }** trata exceções, como caracter digitado no lugar de número.

```
if (e.getSource()==B1)
{   resultado = n1 + n2;   }
if (e.getSource()==B2)
{   resultado = n1 - n2;   }
if (e.getSource()==B3)
{   resultado = n1 * n2;   }
if (e.getSource()==B4)
{ resultado = n1 / n2;   }
if (e.getSource()==B6)
{
int z = (int)n2;
resultado = Math.pow(n1,z);
}
```

Condicionais referentes aos JButtons correspondem às respectivas operações matemáticas


```
    if (e.getSource()==B7)
    { resultado = Math.sqrt(n2); }

    if (e.getSource()==B8)
    { resultado = 1 / n1; }

    if (e.getSource()==B9)
    { resultado = (int) (Math.random()*n2); }

    T3.setText(" "+(float)resultado);
}
} // fim da classe Calculadora2
```

Condicionais referentes aos
JButtons correspondem às
respectivas operações
matemáticas

Exercícios:

21. Compilar e executar a classe Matemática. Verificar os resultados obtidos para a utilização de algumas funções matemáticas do Java.
22. Acrescentar ao aplicativo calculadora as teclas de potência ($\text{numero1}^{\text{numero 2}}$) , raiz ($\sqrt{\text{numero 2}}$) , e $1/\text{numero 1}$, e valor aleatório entre 0 e numero2.
23. Criar aplicativo para calcular as raízes x_1 e x_2 de uma função do segundo grau.

Referências Bibliográficas

- Java 7 - Ensino Didático
Sérgio Furgeri - Editora Érica