



Java - Aula 04

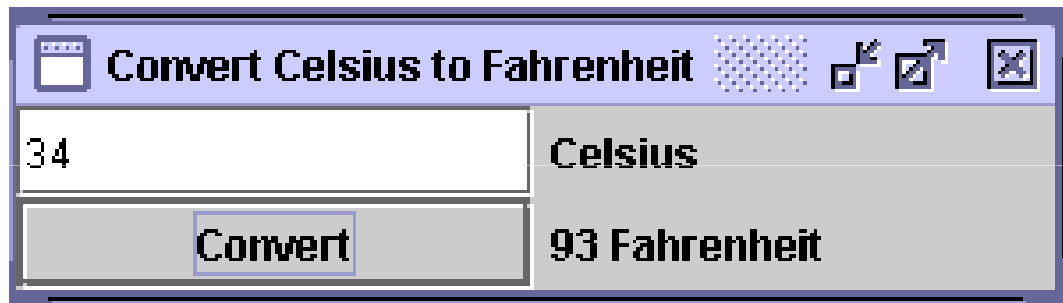
GUI - Swing
MDI

05/09/2012

Celso Olivete Júnior

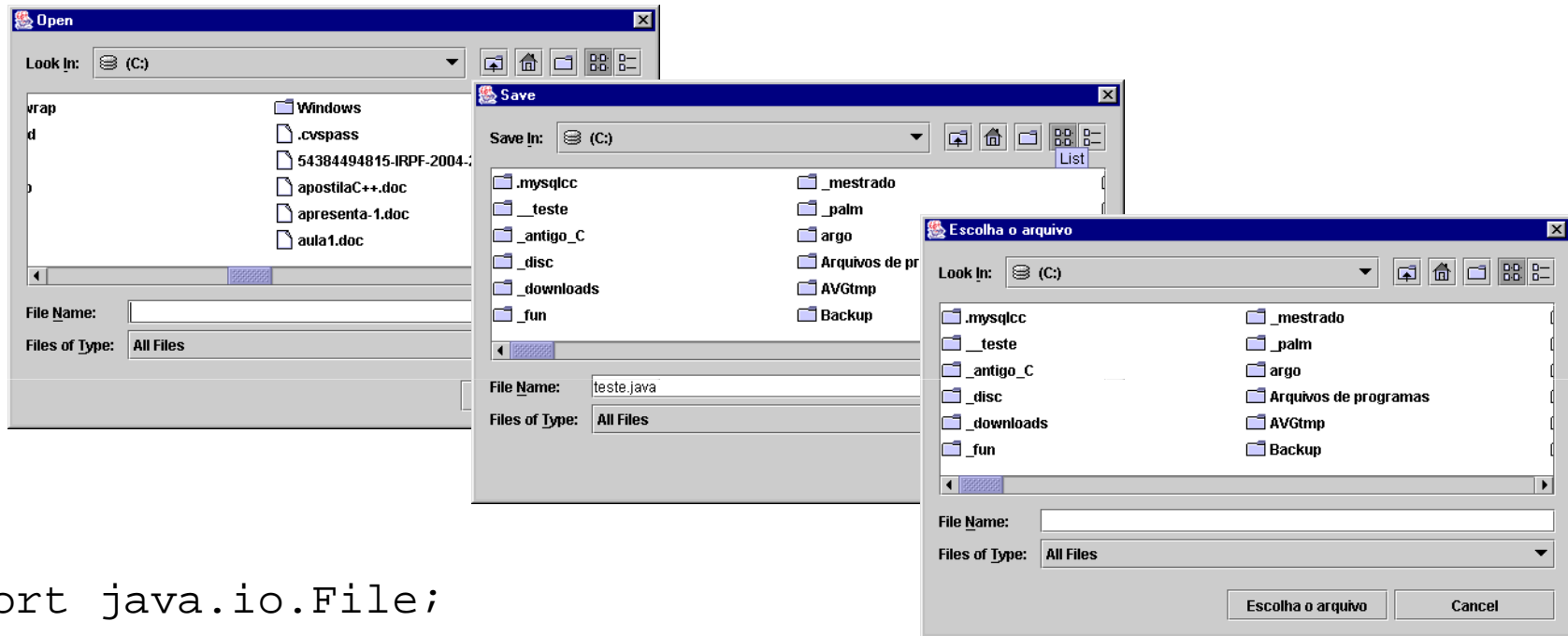
olivete@fct.unesp.br

Exercício para aquecimento...utilize o pacote Swing



$$C = \frac{(F-32)*5}{9}$$

Dialog Windows (Swing)



```
import java.io.File;
...
JFileChooser fc = new JFileChooser("c:\\");
fc.showOpenDialog(this);
File myFile = fc.getSelectedFile();
System.out.println(myFile.getName());
```



FileFilter

- Utilizada para a implementação de filtros que permitem restringir os tipos de arquivos exibidos em um diálogo de seleção
 - FileFilter é uma classe abstrata
- A subclasse de FileFilter deverá implementar os métodos:
 - `boolean accept(File file)`
 - `String getDescription()`



FileFilter

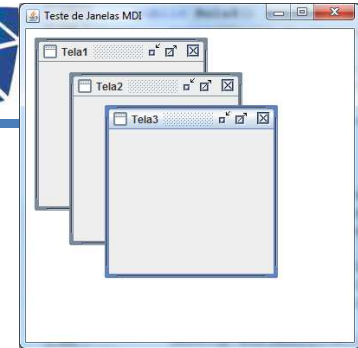
```
import javax.swing.filechooser.FileFilter;

public class GifFilter extends FileFilter
{
    public boolean accept(File f)
    {
        return f.getName().toLowerCase().endsWith(".gif")
            || f.isDirectory();
    }
    public String getDescription()
    {
        return "Arquivos GIF";
    }
}
```



Agora com filtro

```
JFileChooser fc = new JFileChooser("c:\\");
    int result;
    fc.addChoosableFileFilter(new GifFilter());
    result = fc.showOpenDialog(this);
    if (result == javax.swing.JFileChooser.APPROVE_OPTION)
    {
        File myFile = fc.getSelectedFile();
        String fullfileName = fc.getSelectedFile().getAbsolutePath();
        JOptionPane.showMessageDialog(this,
            fullfileName,
            "Nome completo do arquivo selecionado",
            JOptionPane.WARNING_MESSAGE);
    }
```



Aplicações MDI

- **MDI: Multiple Document Interface**

- Programa tem uma janela de "desktop" que contém as demais janelas.
- As janelas filhas podem ser minimizadas e movidas dentro do desktop, mas não removidas.
- Se o desktop for minimizado, todas as janelas filhas também o são.

- **Suporte para MDI no Swing**

- **JDesktopPane**

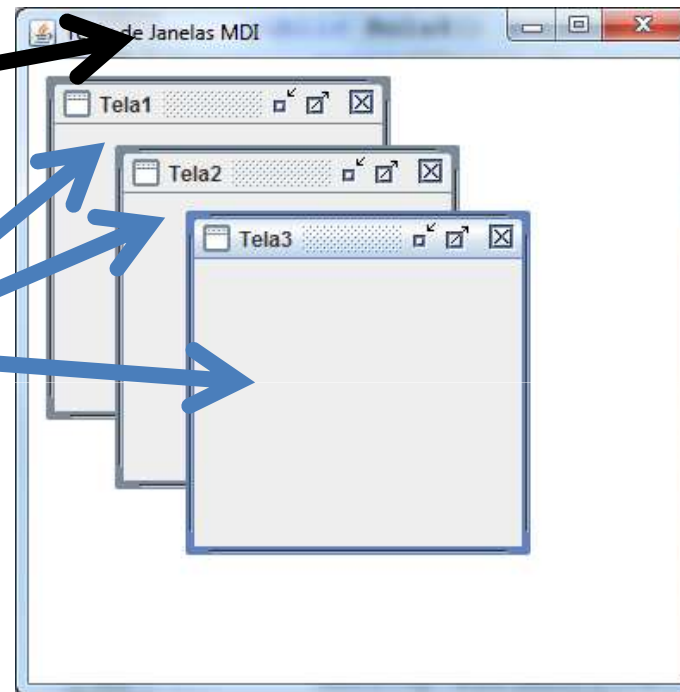
- Serve como a janela mãe (desktop) das demais janelas.

- **JInternalFrame**

- Semelhante a um frame, exceto que ele está restringido a permanecer dentro do objeto JDesktopPane.

Aplicações MDI

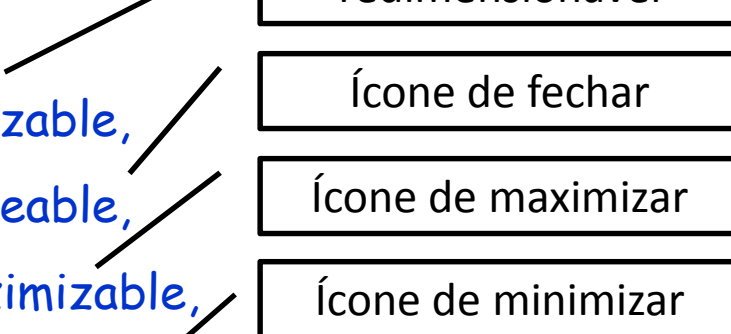
- JDesktopPane
- JInternalFrame



JInternalFrame

- É um contêiner para ser utilizado em aplicações MDI.
- Construtor principal

```
public JInternalFrame(String title,  
                      boolean resizable,  
                      boolean closeable,  
                      boolean maximizable,  
                      boolean iconifiable)
```



- redimensionável
- Ícone de fechar
- Ícone de maximizar
- Ícone de minimizar

- Outros construtores úteis

`JInternalFrame()`

`JInternalFrame(String title)`

`JInternalFrame(String title, boolean resizable)`

`JInternalFrame(String title, boolean resizable, boolean closable)`



JInternalFrame

- Outros métodos:
 - `moveToFront()`
 - `moveToBack()`
 - `setLocation (int x, int y)`
 - `setSize(int largura, int comprimento);`
 - `setMinimumSize(new Dimension(largura, comprimento));`
 - `setTitle(String);`
 - `setMaximizable(boolean);`
 - `setIconifiable(boolean);`
 - `setClosable(boolean);`
 - `setResizable(boolean);`
 - `setJMenuBar(JMenuBar);`



JDesktopPane

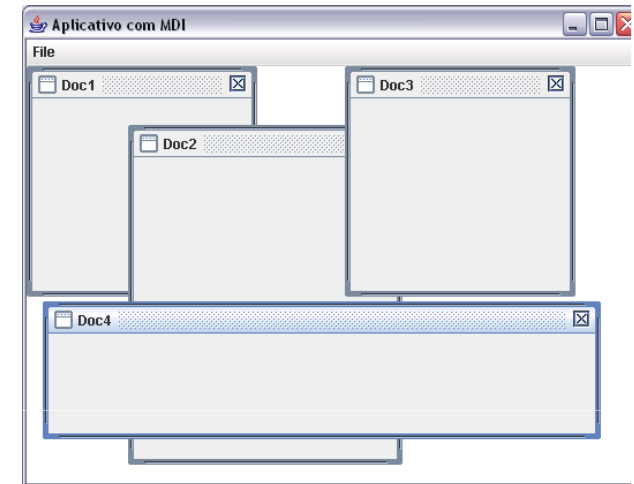
- Um objeto `JDesktopPane` é um contêiner para construir uma aplicação de múltiplos documento (MDI). Serve como a janela mãe (desktop) das demais janelas.
- Um objeto `JDesktopPane` pode conter diversos objetos `JInternalFrame`.
- Principais métodos
 - `add(JInternalFrame)`
 - `cascadeFrames()`



Pacote Swing MDIs (janelas de documento)

```
// Deve-se criar o frame principal com um painel JDesktopPane e as janelas MDI
// devem ser do tipo JInternalFrame
```

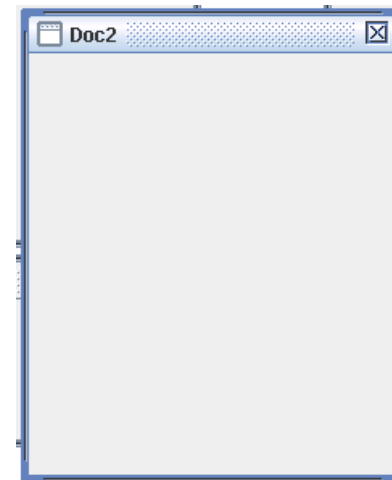
```
class MDIwindow extends JInternalFrame
{
    ...
}
class MDIapp extends JFrame implements ActionListener
{
    JDesktopPane pMDI;
    public MDIapp()
    {
        pMDI=new JDesktopPane();
        getContentPane().add(pMDI);
        ...
    }
    public void actionPerformed ( ActionEvent event )
    {
        ...
        MDIwindow doc=new MDIwindow(); // criação de uma janela MDI
        pMDI.add(doc);
        doc.setVisible(true);
        ...
    }
}
public class MDI extends JFrame
{
    public static void main(String[] args)
    {
        JFrame.setDefaultLookAndFeelDecorated(true);
        MDIapp frame = new MDIapp(); frame.setSize(800, 600);
        frame.setVisible(true);
    }
}
```





Pacote Swing MDIs (janelas de documento)

```
class MDIwindow extends JInternalFrame
{
    private static int MDIcount=1; // contador estático para compor o
    título da janela: Doc1, Doc2,...
    public MDIwindow()
    {
        this.setTitle("Doc"+MDIcount);
        this.setClosable(true); // permite o fechamento
        this.setResizable(true); // permite o redimensionamento
        setSize(200,200);
        MDIcount++;
    }
}
```





Pacote Swing MDIs (janelas de documento)

```
class MDIapp extends JFrame implements ActionListener
{
    JDesktopPane pMDI;
    private JMenuItem New,Close,CloseAll;
    public MDIapp()
    {
        // criação de um menu com as opções "Nova janela", "fechar janelas" e "fechar todas as janelas"
        ...
        pMDI=new JDesktopPane();
        getContentPane().add(pMDI);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
    public void actionPerformed ( ActionEvent event )
    {
        if (event.getSource()== New) // Nova janela
        {
            MDIwindow doc=new MDIwindow();
            pMDI.add(doc); doc.setVisible(true); }
        if (event.getSource()== Close) // fechar janela ativa
        {
            if(pMDI.getSelectedFrame()!=null) pMDI.getSelectedFrame().dispose(); }
        if (event.getSource()== CloseAll) // fechar todas as janelas
        {
            JInternalFrame docs[]=pMDI.getAllFrames();
            for(int i=0;i<docs.length;i++)
                docs[i].dispose(); }
        }
    }
}
```



// criação de um menu com as opções

```
//itens do menu
1.      New=new JMenuItem("New document");
2.      Close=new JMenuItem("Close document");
3.      CloseAll=new JMenuItem("Close all documents");
4.      //registrando os handlers
5.      New.addActionListener(this);
6.      Close.addActionListener(this);
7.      CloseAll.addActionListener(this);
8.      //criando os itens de File
9.      JMenuItem fileMenu = new JMenuItem("File");
10.     fileMenu.add(New);
11.     fileMenu.add(Close);
12.     fileMenu.add(CloseAll);
13.     //criando a barra de menu
14.     JMenuBar bar = new JMenuBar();
15.     setJMenuBar( bar );
16.     bar.add(fileMenu);
```



Abrindo e mostrando uma imagem

```
try {
    BufferedImage image = ImageIO.read(myFile);
    String imageInfo = "Dimensions: "+image.getWidth()+"x"+
        image.getHeight()+" Bands: "+
        image.getRaster().getNumBands();
    ImageIcon icon = new ImageIcon(image);
    JLabel imageLabel = new JLabel(icon);
    MDIwindow frame = new MDIwindow();
    frame.setTitle("Imagem apresentada: "+fullfileName);
    frame.getContentPane().setLayout(new BorderLayout());
    frame.getContentPane().add(new
JScrollPane(imageLabel),BorderLayout.CENTER);
    frame.getContentPane().add(new JLabel(imageInfo),
    BorderLayout.SOUTH);
    frame.setDefaultCloseOperation(DISPOSE_ON_CLOSE);
    frame.setSize(400,400);
    this.getContentPane().add(frame);
    frame.setVisible(true);
}
catch(Exception e){}
```




Mostrando em tons de cinza

```
try {
    BufferedImage image = ImageIO.read(myFile);
    int k;
    int width = image.getWidth();
    int height = image.getHeight();
    int nbands = image.getSampleModel().getNumBands();
    for(int h=0;h<height;h++)
        for(int w=0;w<width;w++)
            {
                int rgb = image.getRGB(w,h);
                int r = (int)((rgb&0x00FF0000)>>>16); // Red level
                int g = (int)((rgb&0x0000FF00)>>>8); // Green level
                int b = (int) (rgb&0x000000FF); // Blue level
                k = r+g+b;
                k=k/3;
                k=k+(int)((k<<8));
                k=k+(int)((k<<16));
                image.setRGB(w,h,k);
            }

    String imageInfo = "Dimensions: "+image.getWidth()+"x"+ . . .
```



Trabalho: Paint - **Data de entrega: dia da Prova**

- apresentar as principais formas geométricas (Polígono, retângulo, reta, círculo, elipse, ponto, mão livre,...)
- desenho com o mouse (cada uma das formas deve ser desenhada a partir das coordenadas do mouse.
- utilizar múltiplas janelas (MDI)
- opções de abrir e salvar uma figura (componente JFileChooser) - utilize o banco de dados MySql