

Camada de apresentação =====

A camada de apresentação de um sistema de computador é a parte do software responsável em interagir com o usuário. Muitas vezes chamadas de "Telas" do sistema, a camada de apresentação deve ser construída separadamente do código da camada de domínio do problema, facilitando a utilização do mesmo software em diferentes tipos de equipamentos (Ex.PCs, Handhelds, Celulares, etc...).

Java oferece uma biblioteca rica em recursos para a construção de interfaces gráficas para usuários (GUI - Graphical User Interface).

Para iniciarmos esta abordagem, vamos utilizar a classe **JFrame** que permite criar uma janela padrão para aplicações.

Uma boa prática durante a utilização do JFrame, é criar uma classe que estenda seus recursos, facilitando assim a sua particularização. Exemplo:

```
public class FormPrincipal extends JFrame {  
    public FormPrincipal() {  
        setTitle("Janela principal");  
        setSize(300,200);  
    }  
}
```

O código acima define uma classe que herda as características e métodos de um **JFrame**. Dentro do construtor, usamos o método **setTitle** herdado de JFrame que define o título a ser apresentado na janela. O método **setSize** (também herdado de JFrame) define o tamanho (largura, altura) da janela em pixels.

Sendo assim, podemos utilizar o código de uma aplicação Java abaixo (método main) para apresentá-la.

```
public class PrincipalJFrame {  
    public static void main(String[] args) {  
        FormPrincipal janela = new FormPrincipal();  
        janela.setVisible(true);  
    }  
}
```

O método **setVisible(true)** (herdado de JFrame), torna a janela visível. A classe JFrame possui vários outros recursos que podem ser utilizados (consultar a documentação da API do Java). Destacamos alguns deles a seguir:

setDefaultCloseOperation

Define que operação deve ser realizada ao pressionarmos o botão de fechamento da janela (☒). Uma boa opção neste momento é utilizar a constante `EXIT_ON_CLOSE`, que indica que a aplicação deve ser encerrada ao pressionar tal botão. Exemplo:

```
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

setLocation

Permite definir a posição (coluna, linha) onde será apresentada a janela. Caso não seja especificada, a janela é apresentada na coluna 0 (zero) linha 0 (zero). Exemplo:

```
setLocation(200,100);
```

Como descobrir o tamanho da tela

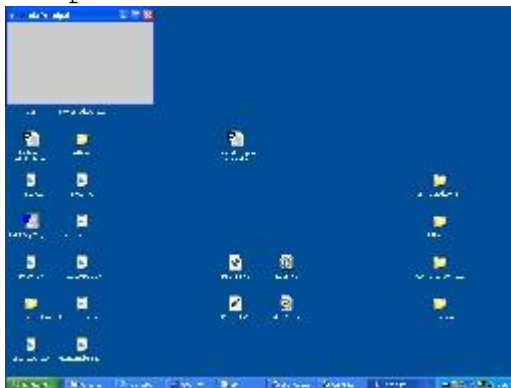
A biblioteca Java possui uma classe chamada `Toolkit` que oferece vários métodos para a manipulação dos recursos existentes no ambiente de execução. Segue abaixo um trecho de código que permite sabermos a resolução de vídeo configurada no sistema operacional:

```
Toolkit ferramenta = Toolkit.getDefaultToolkit();  
Dimension tamanhoTela = ferramenta.getScreenSize();  
int altura=tamanhoTela.height;  
int largura=tamanhoTela.width;
```

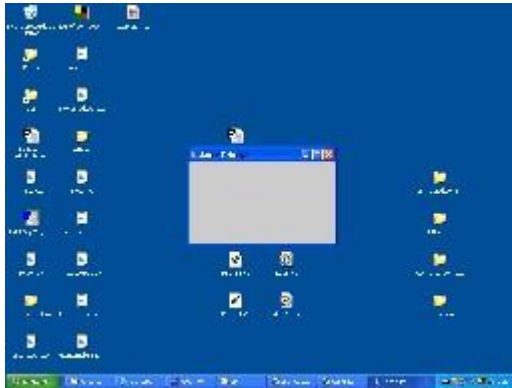
Exercícios

- 1) Crie um projeto chamado `Java2_0003_JFrame_Ex1` que apresente uma janela de 300x200 pixels no canto superior esquerdo da tela.

Exemplo:



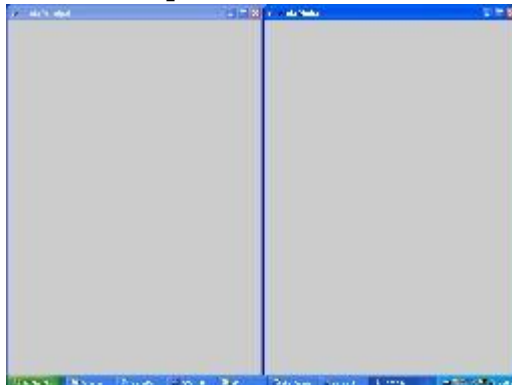
- 2) Crie um projeto chamado Java2_0003_JFrame_Ex2 que apresente uma janela de 300x200 pixels centralizada na tela. Exemplo:



- 3) Crie um projeto chamado Java2_0003_JFrame_Ex3 que apresente duas janelas de 300x200 pixels centralizadas em cada metade horizontal da tela. Exemplo:



- 4) Crie um projeto chamado Java2_0003_JFrame_Ex4 que apresente lado a lado, duas janelas, cada uma de tamanho correspondente à metade da tela. Exemplo:



- 5) Fechar o Eclipse, e executar os programas acima através do Prompt de comando.
6) Visualizar o exemplo "SwingSet2.html" disponível na pasta "\demo\jfc\SwingSet2" da instalação do Java.

Final do documento

Prof.Fernando J.C.Branquinho

<http://br.geocities.com/brangs>