**Eduardo Sinico, Ezequiel & Matheus - *SENAI* – SA3 LayoutManager**

**Relatório das Calculadoras em JAVA**

Este é um relatório demostrando como foi feito o layout da interface da calculadora de Programador em Java Swing:

Em Java, "layout" se refere à maneira como os componentes gráficos de uma interface de usuário são organizados em um contêiner, como uma janela ou um painel. O layout determina a disposição e o posicionamento dos elementos visuais em relação uns aos outros. Layouts são essenciais para criar interfaces de usuário atraentes, responsivos e funcionais em aplicativos Java.

**Tópicos:**

* Calculadora de Programador;

- BorderLayout;

- GridLayout;

- FlowLayout.

* Calculadora de IMC;
* Calculadora Padrão.
* **Calculadora de Programador:**

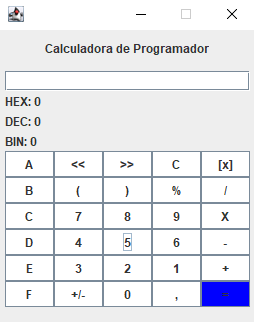
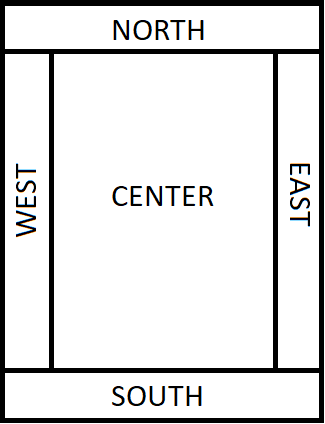


Imagem da Calculadora de Programador

Esses são os três tipos de layout usado no programa:

* **BorderLayout**: BorderLayout é um layout pré-determinado em Java onde ele define as posições de norte, sul, leste, oeste e o centro da página.



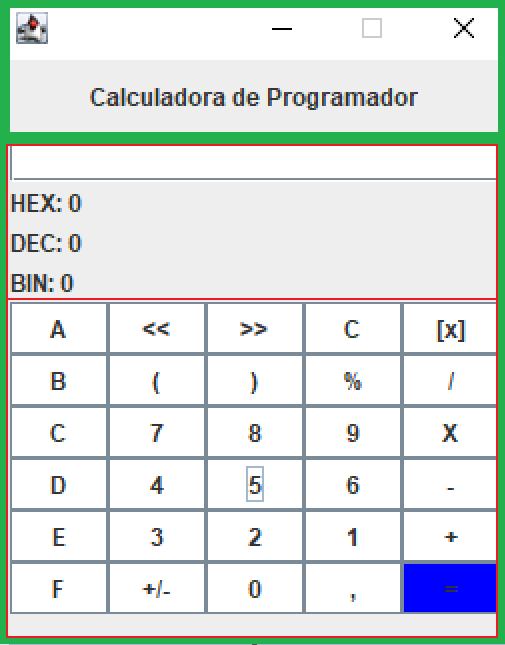
Exemplo de BorderLayout

**No projeto:**

Primeiramente eu criei um JFrame e adicionei o layout dele para BorderLayout, após isso, adicione dois painéis com suas determinadas posições.

*JFrame* janela **= new** JFrame();

janela.setLayout(**new** BorderLayout());



Exemplo da utilização de BorderLayout na calculadora

No projeto de Java, as cores verde e vermelha representam onde está localizado o BorderLayout. Começando pela parte verde, eu defini dois painéis, uma para o norte:

*JPanel* painelTop **= new** JPanel();

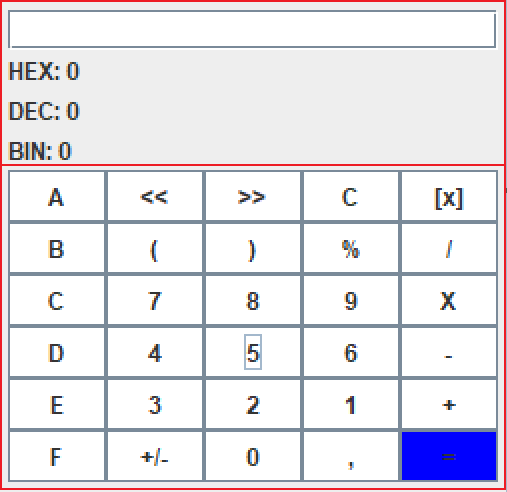
janela.add(painelTop,BorderLayout.NORTH);

E outra para o centro:

*JPanel* painelCenter **= new** JPanel();

janela.add(painelCenter,BorderLayout.CENTER);

Indo para o a parte vermelha, eu criei um segundo painel principal, com a mesma configuração de Layout para o JFrame, o BorderLayout:



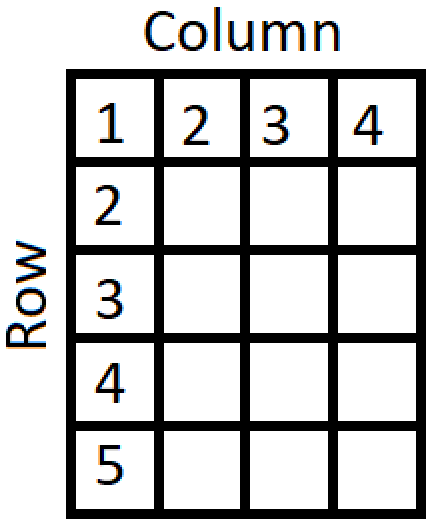
Neste segundo painel principal, eu criei dois painéis e alinhei eles um de baixo do outro. O TextField do resultado, e os valores HEX, DEC e BIN estão posicionados na parte superior, enquanto, as teclas da calculadora estão posicionadas na parte inferior, utilizando a mesma lógica nos processos anteriores.

* **GridLayout**: GridLayout é outro estilo de Layout bem utilizado em Java. Nela, podemos definir Painéis ou Frames com determinadas quantidades de linhas e colunas. Geralmente as JFrames criadas em Java já possuem este Layout.

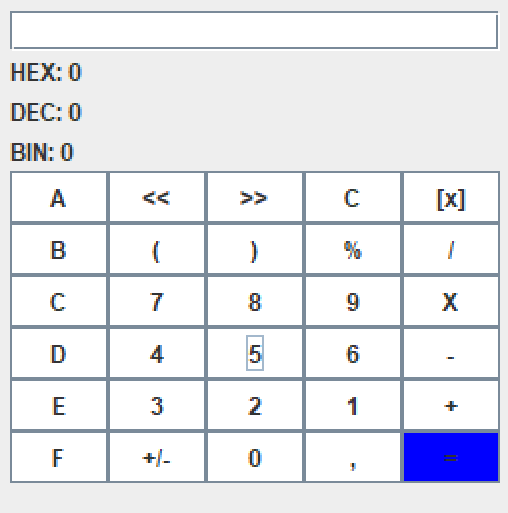
**Funcionalidade do GridLayout:**

*JPanel* painel **= new** JPanel();

painel.setLayout(**new** GridLayout(rows: 5, cols: 4));

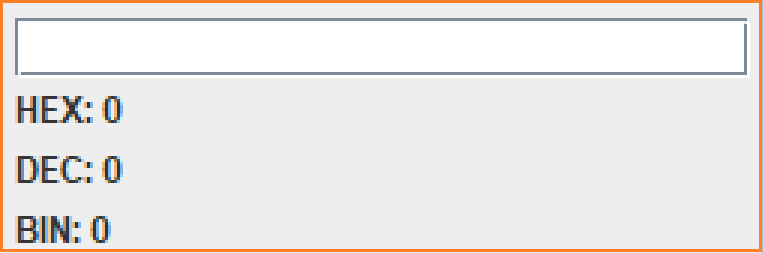
****

**No projeto:**

****

No Layout da calculadora, foi utilizado o ***GridLayout*** em dois painéis.

O primeiro painel apresenta um GridLayout com 4 linhas e uma coluna:



**Código:**

painelResults.setLayout(**new** GridLayout(rows: 4, cols: 1));

painelResults.add(**new** JTextField());

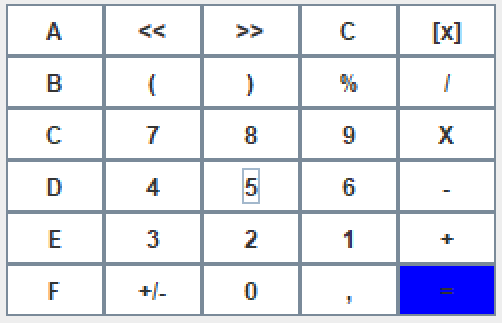
painelResults.add (**new** JLabel("HEX: " + 0));

painelResults.add(**new** JLabel("DEC: " + 0));

painelResults.add(**new** JLabel("BIN: " + 0));

**OBS.:** É necessária uma coluna para que os elementos possam estar um debaixo do outro.

Os mesmos procedimentos de ***GridLayout*** foram aplicados para o segundo painel, onde apresenta os botões da calculadora:

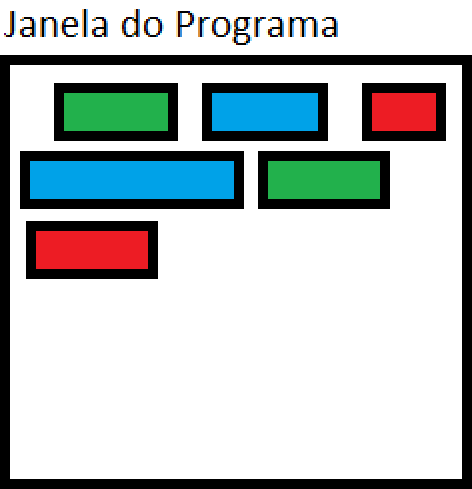


**Código:**

painelCalc.setLayout(**new** GridLayout(rows: 6, cols: 5));

Percebe-se que a “tabela” de botões apresenta 6 linhas e 5 colunas, onde destes 30 elementos, cada um ocupa um espaço desta lista, seguindo a ordem da esquerda para direita e de cima para baixo.

* **FlowLayout**: Por fim, mas não menos importante, o FlowLayout é mais um Layout pronto na Linguagem Java. Ele alinha os elementos um ao lado do outro até o fim da janela, caso não haja espaço, ele pula para parte de baixo do programa.



**No projeto:**



No projeto, a única parte que possuí o Layout Flow está no próprio título. Entretanto, não foi necessário atribuir este Layout para este painel, pois ela já vem atribuída automaticamente para os JPanels. Mas você pode seguir o seguinte código:

painel.setLayout(**new** FlowLayout());

* **Calculadora de IMC:**