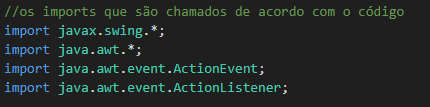
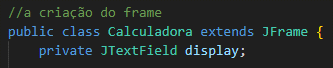
***BORDER LAYOUT EM JAVA***

* *IMPORTS:*



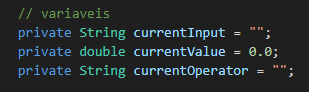
Se trata dos métodos de JAVA, que são chamados de acordo com o código para ocorrer as ações solicitadas.

* *FRAME:*



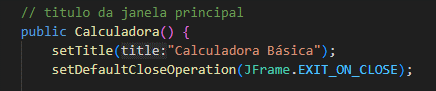
Se trata da criação do frame, que seria a janela principal do código a partir da abertura dos comandos acionados. E o extends estaria puxando o JFrame para a criação da janela que englobaria os painéis.

* *VARIÁVEIS:*



Nada menos do que a criação das variáveis principais dos comandos do JFrame e dos painéis.

* *JANELA PRINCIPAL:*



Aqui se trata do comando, para identificar o nome que você quer que a janela principal tenha e nos mostre a partir do momento que ela é aberta.

-EXIT\_ON\_CLOSE:é o comando para ter a operação pra conseguir fechar a janela principal.

* *PAINÉIS:*



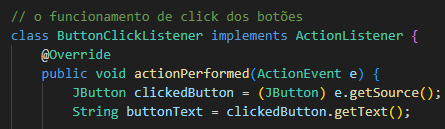
Aqui está o comando que faz a mágica acontecer, o começo da criação do painel principal e do display que vem todas as ações e os componentes.

* *BOTÕES:*



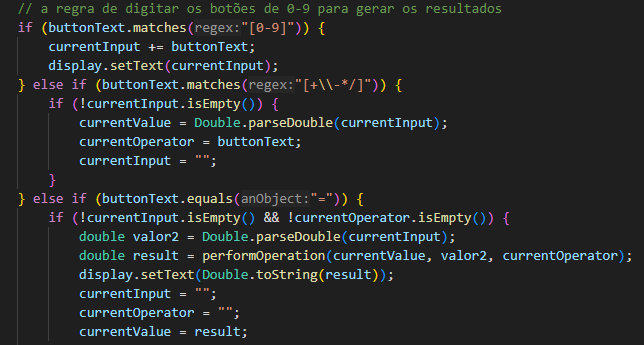
Acima mostra como foi criado os botões, as funções de click e as labels(que seria o escrito que vai vir em cada botão), e a quantidade de linhas e colunas que vai ser dividido.

* *FUNCIONAMENTO DOS BOTÕES:*



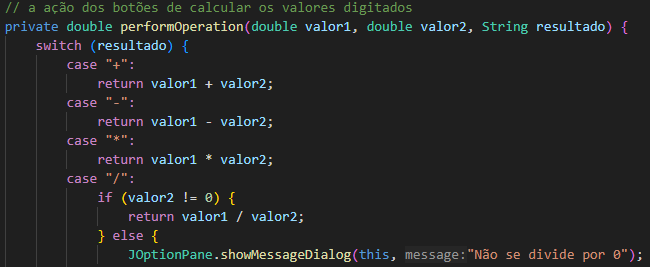
Os comandos dos botões foram colocados com o ActionListener, para acionar a ação de click indicada nos botões.

* *REGRA DOS VALORES:*



Essa parte é onde distinguimos a regra de poder digitar apenas números de 0-9, e pro símbolo = mostrar os resultados dos botões selecionados.

* *CÁLCULO PARA OS RESULTADOS:*



Está é a operação para retornar ao usuário o resultado dos problemas matemáticos básico, com os valores que podem ou não ser divididos.

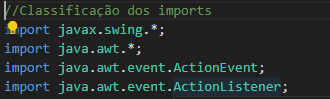
* *RESULTADOS:*



Aqui se trata do final do código que seria a demonstração do resultado, que podem ser mostrados em decimais.

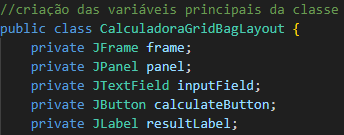
***GRIDBAG LAYOUT EM JAVA***

* *IMPORTS:*

******

Eles mantem os mesmos imports do anterior.

* *VARIÁVEIS PRINCIPAIS:*



Aqui as coisas ficam diferentes, pois, diferente do anterior, as variáveis são:”JFrame”, “JPanel”, “JTextField”, “JButton”, “JLabel”.

JFrame: para ser criada a janela quando executado.

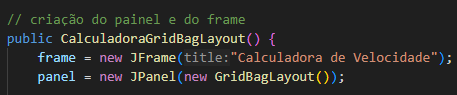
JPanel: Para haver um painel (ou uma div se preferir) que é onde você pode organizar seu componentes que vão aparecer na janela quando executado

* *DEIXANDO A JANELA VISIVEL:*



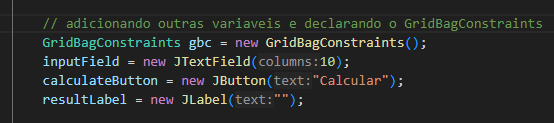
Basicamente esta parte deixando a janela visível, mas por algum motivo quando incrementado “true” na parte da “boolean b” apresentava erros que eram impossíveis de se resolver, caso alguém ou até mesmo o professor saiba, adoraria em saber.

* *DEFININDO O TITULO DA JANELA E LAYOUT:*

**

Nessa parte, estão sendo definidos o layout e o titulo que irá aparece na parte superior da janela, o layout sendo o:”GridBagLayout” que permiti que os componentes sejam posicionados em grades flexíveis.

* *ADICIONANDO OUTRAS VARIAVEIS E O GRIDBAGCONSTRAIS:*



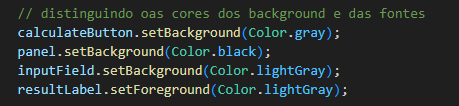
Essa outras variáveis (componentes) são “InputField”, “CalculateButton” e “ResultLabel”

InputField: aqui será o espaço para o texto ser colocado

CalculateButton: aqui será colocado um botão, onde sua função será calcular os valores dos números

ResultLabel: aqui será colocado um texto fixo ou nesse caso, será uma saída de dados para o resultado final ficar visível para o usuario (se eu não me engano.)

* *COLOCANDO CORES NO BACKGROUND E NA FONTE:*



Aqui nós estamos adicionando cores, para chamar mais a atenção do usuário sendo elas as cores:

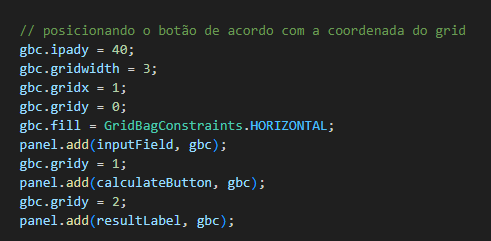
-Cinza

-Preto

-Cinza Claro

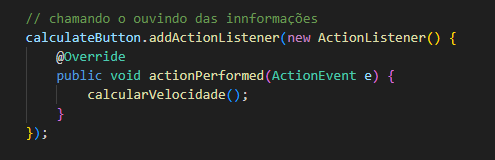
Cada uma em um dos locais mencionados anteriormente, variando entre background e foreground, ou seja, cores na área da frente e de trás

* *POSICIONANDO O BOTÃO:*



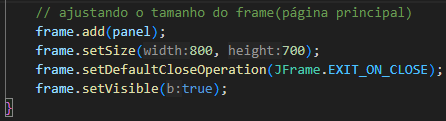
Aqui nós estamos posicionando o botão “Calcular” de acordo com as coordenadas do gridBag e com o auxilio do objeto GridBagConstraints isso se torna mais flexível e facil(é importe dizer que as coordenadas são em pixeis ou seja, os números que estão no exemplo acima são em pixeis com os componentes tendo eixos “X” e “Y”) .

* *ADIONANDO AÇÃO AO BOTÃO “CALCULAR”:*



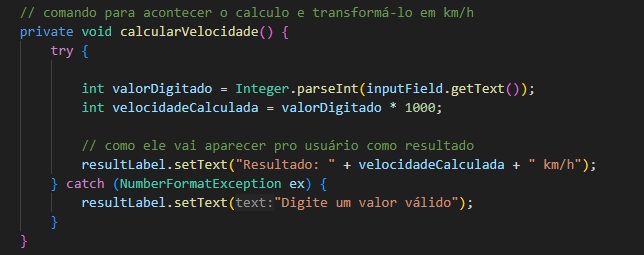
Aqui estamos adicionando uma ação ao botão “Calcular” com o “ActionListener” que basicamente irá calcular o número que você colocar e irá transformar em km/h e o resultado está em uma “public void” para que o resultado não seja salvo e o “Override” está sendo usado para subscrever e implementar um método da interface “ActionListener” o “actionPerformed” que será ele que irá lidar com o “evento” (mais conhecido como “click” do mouse no botão “Calcular”)

* *AJUSTANDO O TAMANHO DA FRAME:*



Aqui estamos adicionando o painel a frame, definindo o seu tamanho em 800 pixeis de largura e 700 pixeis de altura, adicionamos a função “Exit\_on\_close” para o usuário poder fechar a janela e deixar ela visível ao mesmo.

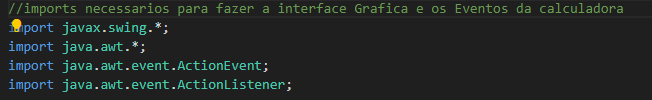
* *FAZENDO O CALCULO:*



E Por fim, aqui nos vamos fazer o calculo do valor digitado e transforma-lo em km/h.

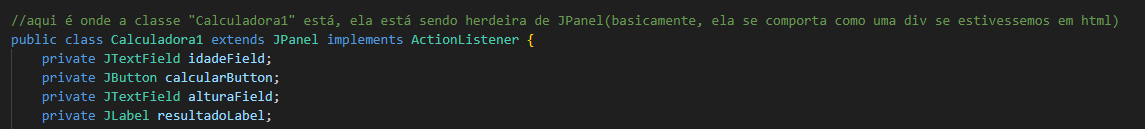
***GRID LAYOUT EM JAVA***

* *IMPORTS:*



Esses imports são necessários para a criação da interface.

* *VARIAVEIS PRINCIPAIS:*



Aqui estou colocando um “JTextField”, um “JButton”, “JLabel” que já foram comentados anteriormente (eu imagino)

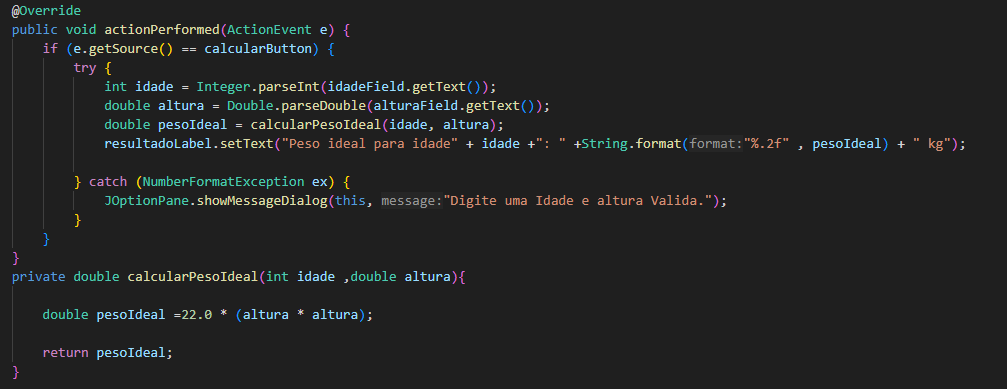
* *RESUMO SOBRE O LAYOUT E A CALCULADORA:*



Aqui estou definindo as linha e colunas junto o layout grid, estou ajustando o tamanho da janel com 300 pixeis de largura e 150 pixeis de altura.

Adicioneis um texto para identicar os espaços necessários e o botão calcular e por fim adicionei todos a frame.

* *RESUMO SOBRE COMO O CALCULO SERÁ FEITO:*



Aqui o “Override” está sendo usado para adicionar a ação ao botão “Calcular” igual anteriormente.

A parte do “if” é para quando a idade do usuário e altura forem digitados, o programa irá retonar o valor do peso ideal para a idade com o “return”.

E a parte do “catch” é para caso o usuário digite por exemplo “idade:1,8” o programa deve retornar essa mensagem para o usuário.