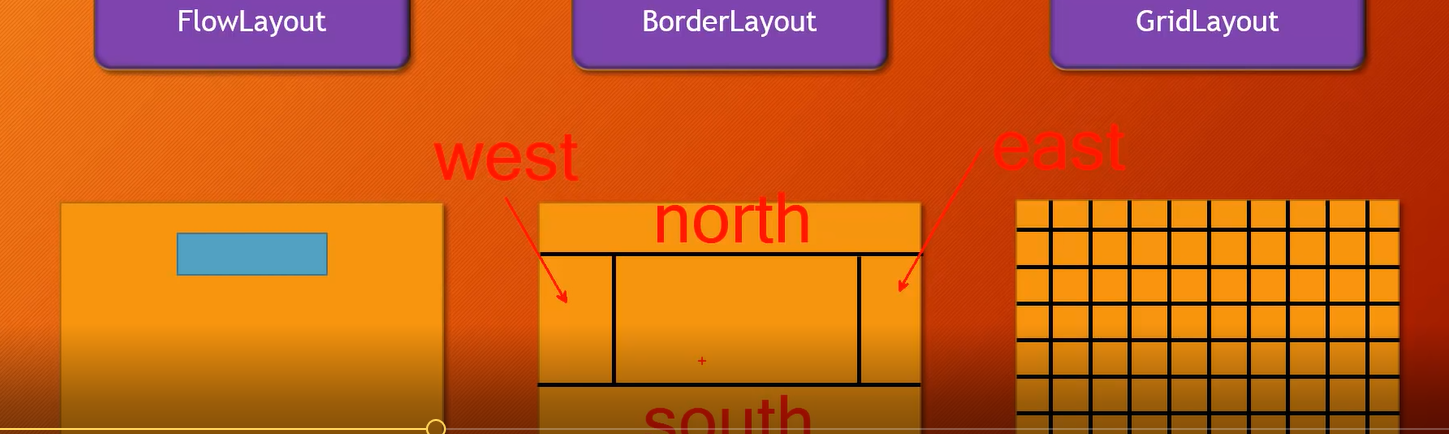
Layouts 12 / 07 / 2017

.

Video 81

Disposiciones o Layouts, (donde colocar los componentes dentro de un marco o ventana o contenedor)

Java ofrece unas disposiciones por defecto. Son las siguientes.

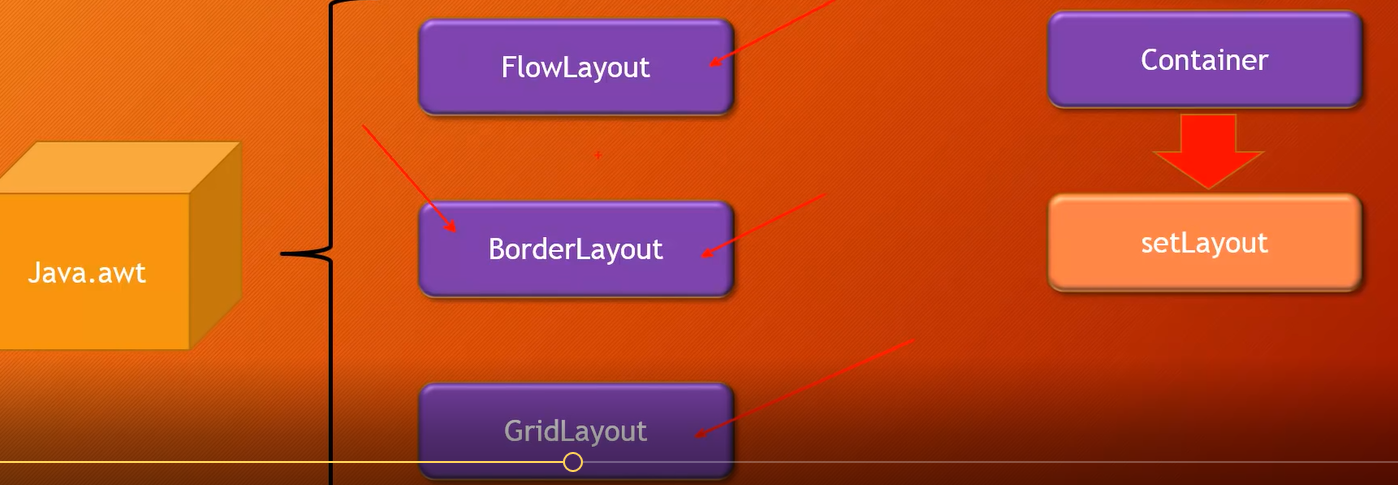


GridLayout \*--divide en celdas y coloca los objetos en las celdas oportunas.

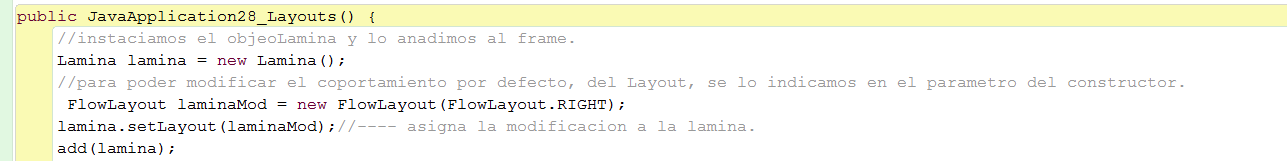
BorderLayout \*--divide el contenedor en las zonas que se ve, colocas los componentes en la zona elegida.

FlowLayout \*--disposicion que utiliza por defecto las laminas.

SI QUIERO UTILIZAR UNA DETERMINA Layout o Disposicion tengo que utilizar su clase correspondiente. El mt, setLayout() que pertence a la cl Container de la que hereda JPanel, luego dentro de Jpanel se puede usar.



Antes de añadir la lamina al marco, modificamos su posicionamiento, recordemos que para laminas, Java utiliza por defecto ‘FowLayout’.

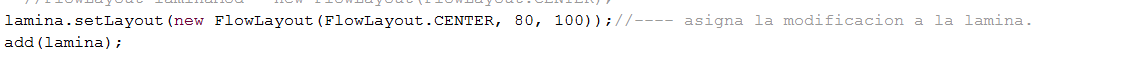


Video 82

Modifica la separación entre objetos que java dá por defecto. Para ello nos vamos a la api y vemos los costructores

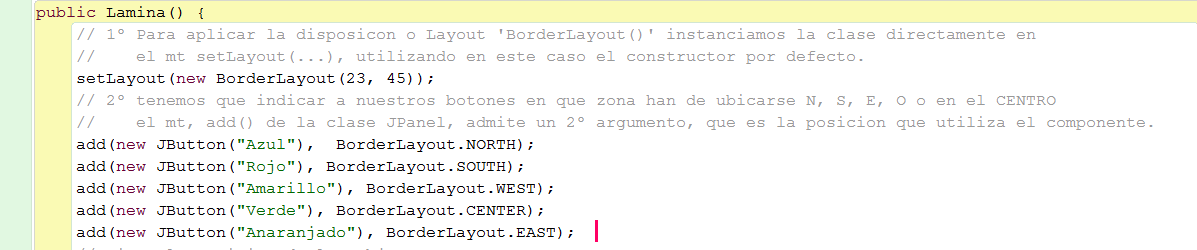


Lo utilizamos.



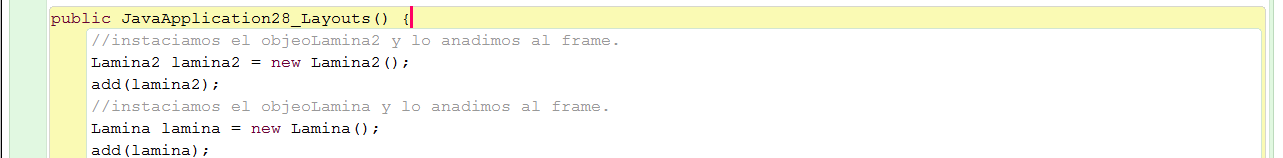
UTILIZACIÓN DE BorderLayout

BorderLayout \*--divide el contenedor en las zonas que se ve, colocas los componentes en la zona elegida. La caracteristica princpal es que el componente se hace igual de grade que la zona. Mn 5 , 49



APLICACIÓN DE DOS TIPOS DE LAYOUT A UN MISMO MARCO.

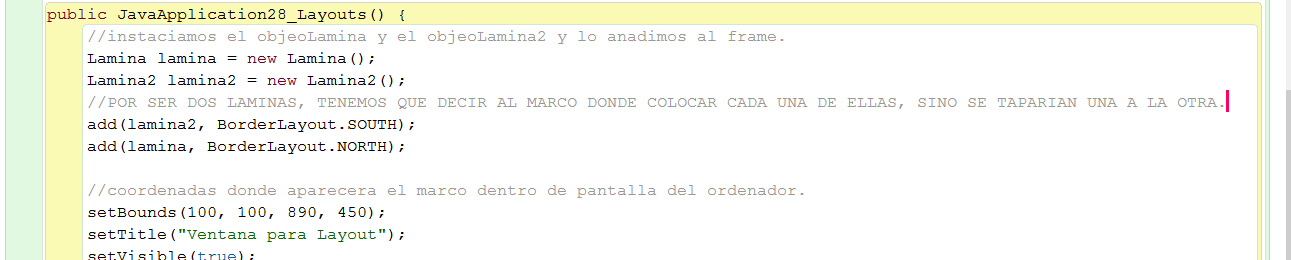
tenemos que crear dos láminas, una para cada ‘disposición’ o ‘Layout’, y agregar ambas al marco.



Pero esto no funciona porque una lamina machacaría a la otra y una de ellas no se vería.

Para solucionarlo, tenemos que decir al marco la disposición que tendrá cada una de las láminas, **RECORDEMOS** que al mt add(…, …), se le pueden pasar dos parámetros.

----le decimos que la 1º lámina la ponga arriba y Lamina2 abajo



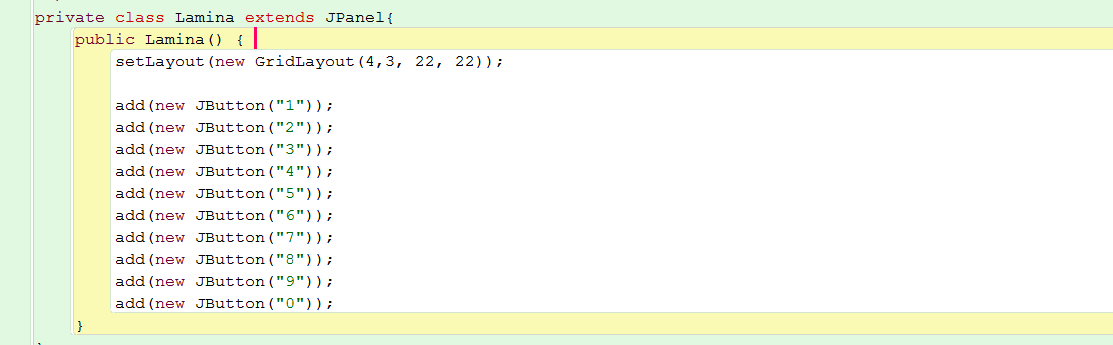
.

.

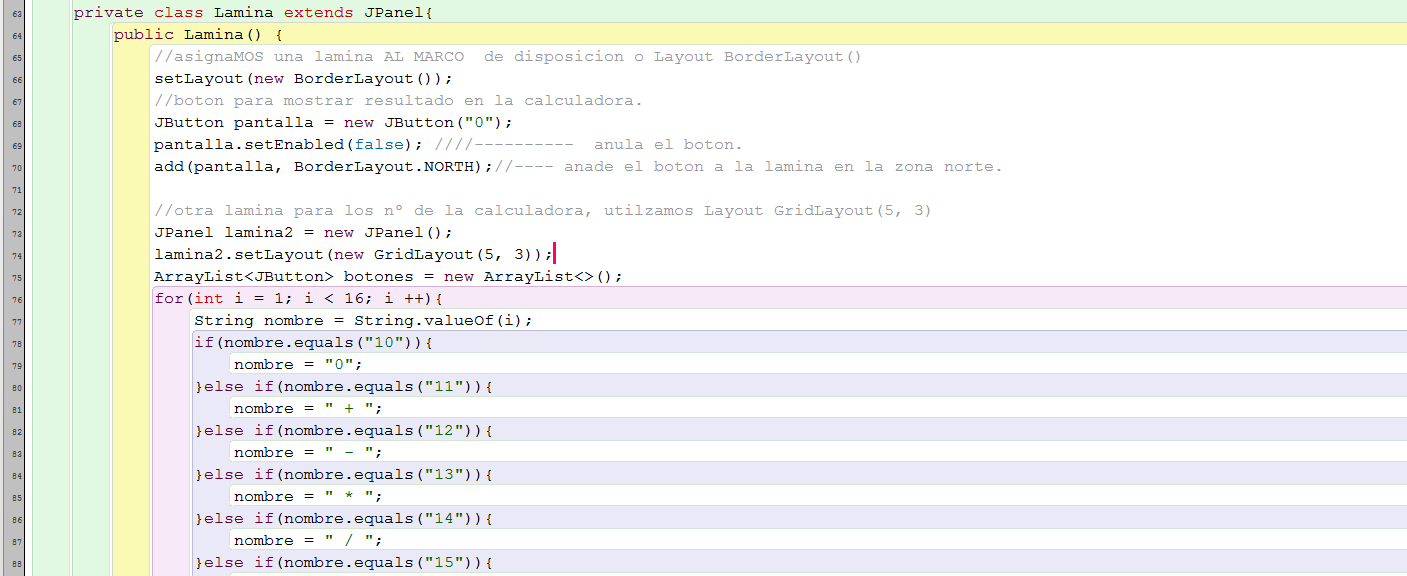
Video 83

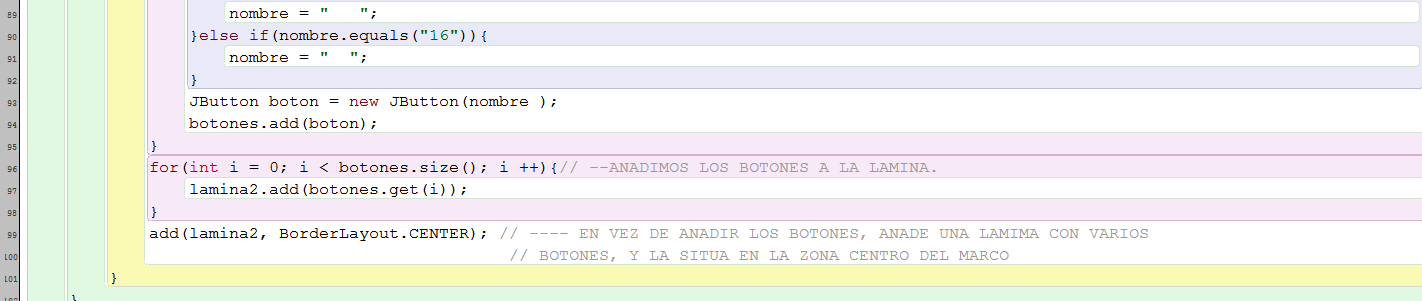
.DISPOSICION GridLayout

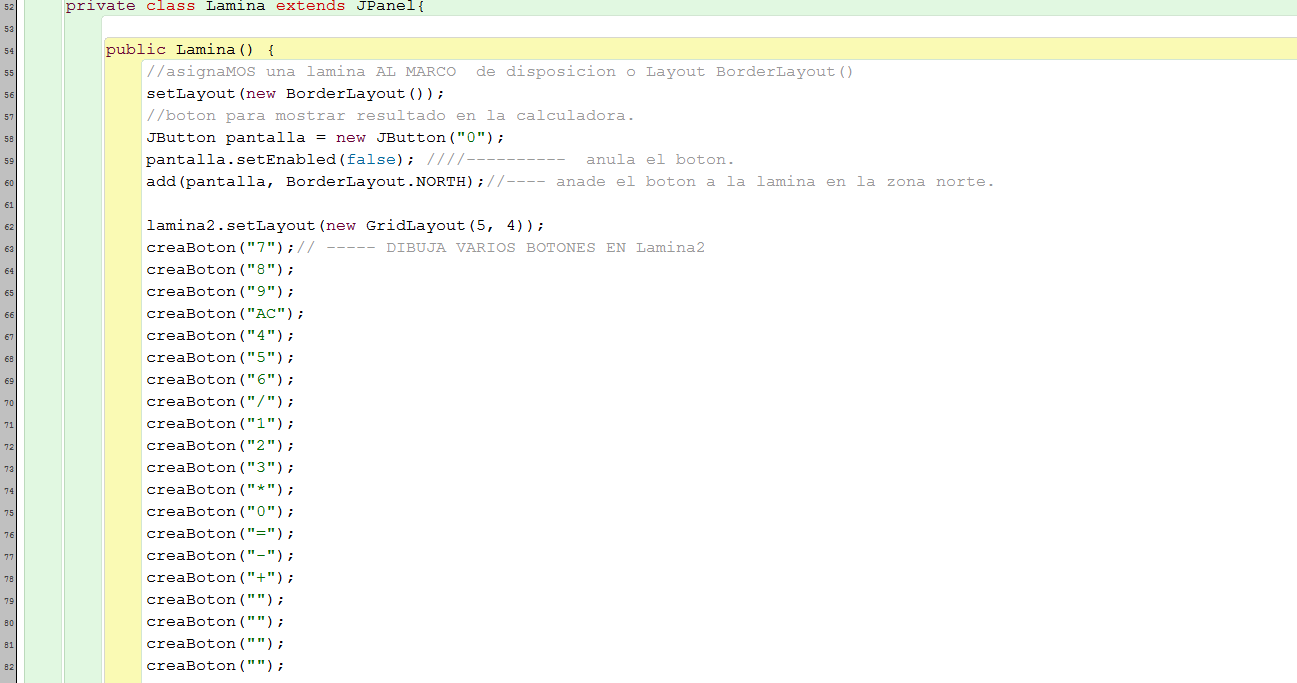
Lo que hace es dividir el contenedor en celdas.



APROVECHAMOS ESTA DISPOSICIÓN PARA CREAR UNA CALCULADORA.



.R OTRA FORMA DE CODIFICARLO





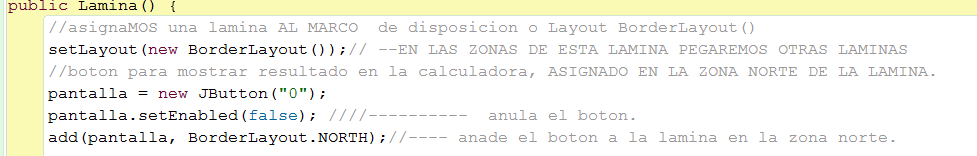
.

Video 84

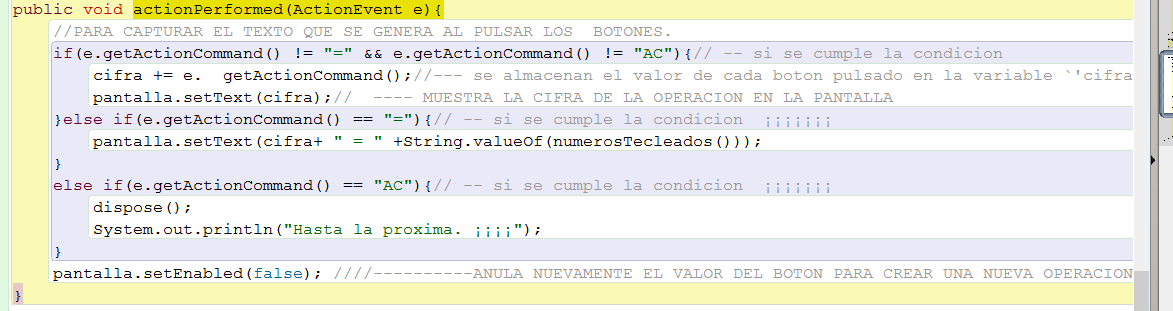
.DAMOS FUNCIONALIDAD A LOS BOTONES DE LA CALCULADORA.

1º PARA QUE APARECCAN LOS Nº TECLEADOS EN LA PARTE SUPERIOR DE LA CALCULADORA

Recordemos que en la parte superior de la pantalla hemos colocado un boton, llamado ‘pantalla’



Los nº tecleados serían el texto asignado al boton en el mt actionPerformed(ActionEvent e){

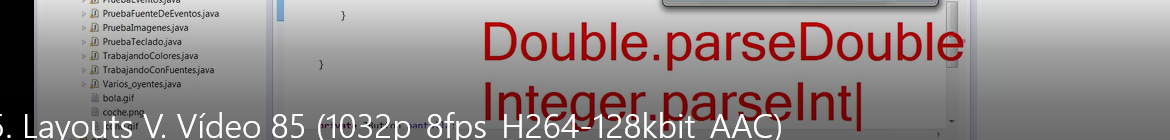


Video 85

Codigo para las operaciones de la calculadora, suma, resta, división etc.

Mn 9

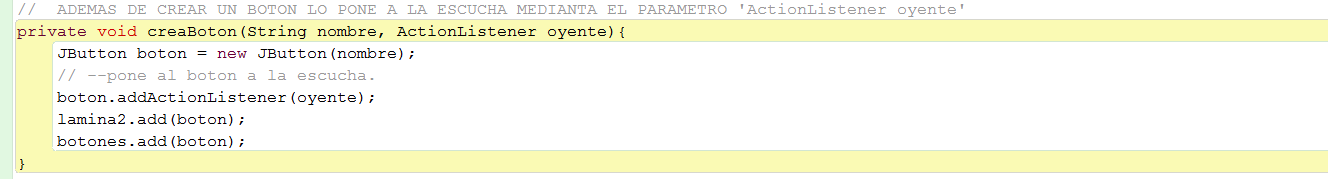
Para pasar a double y para pasar a entero.



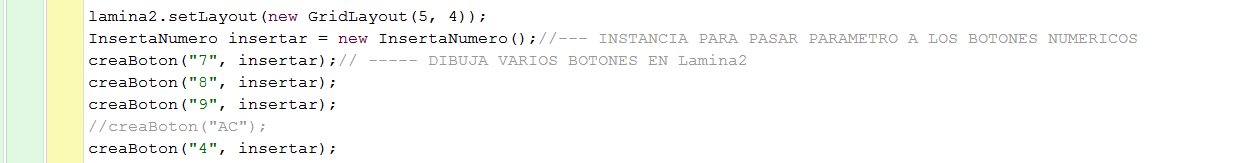
Para pasar a double el String que aparece en panttalla

Calcular(Double.parseDouble(pantalla.getText()));

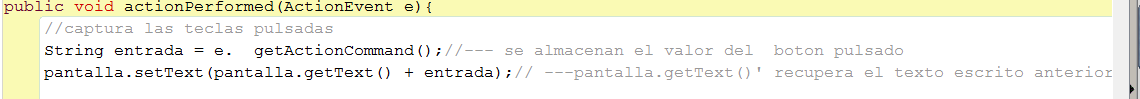
**OTRA MANERA DE CODIFICAR LA CALCULADORA.**



2º AÑADIMOS LOS BOTONES NUMERICOS A LA LAMINA



3º MUESTRA EL VALOR DE LOS BOTONES EN LA PANTALLA DE LA CALCULADORA.



4º ---- ADD FUNCIONALIDAD A LOS BOTONES, para ello 1º anadimos los botones de las operaciones y creamos una variable que almacene los nº tecleados, el resultado de las operaciones.

Y una 2º variable que almacene las operaciones realizadas, +, - , \*, /

Y una 3º variable que almacene la última operacion realizada, por el usuario para si es un ‘=’ muestre el resultado.

--

..

.

.

.