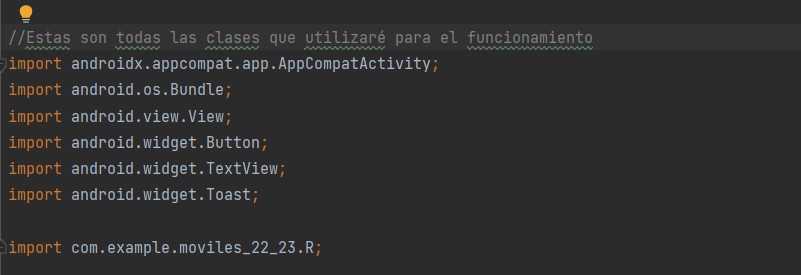
Calculadora

Funcionamiento

Lo primero de todo. Importamos todas las clases que vayamos a utilizar:



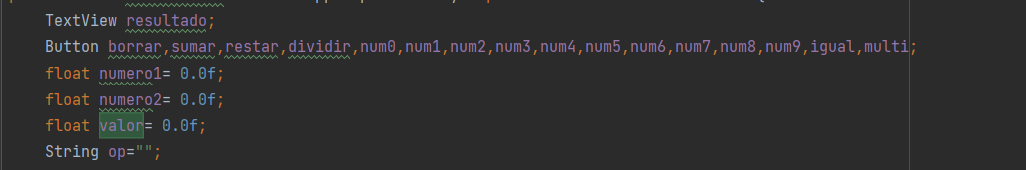
En mi clase principal yo elijo importar la vista OnClickListener.

Con eso tengo 1 sólo listener para todos los botones.

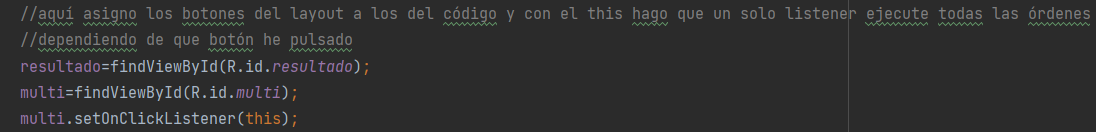
La alternativa es que para cada botón hago un clicklistener y crear 30 lineas de código para cada botón.



Antes del OnCreate, creo todas las variables que vaya a usar en mi código. Así no necesito una variable diferente para cada método, las puedo usar directamente de arriba.



A continuación, digo a mi programa que las variables que voy a usar en el código son las mismas que las del layout.



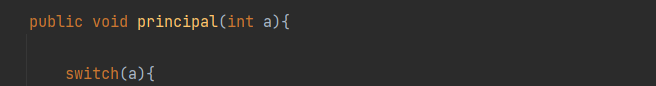
Eso se hace con el findViewById(R.id.”nombre de la variable del layout”)

El setOnClickListener sirve para que interprete que cada botón tiene una orden diferente. Eso lo explicaré más adelante.

Cada botón tiene su id del layout y su respectivo setOnClickListener



Nos adentramos en el corazón del código y la magia del this.



Con un método que recibe un número y no devuelve nada, creo un switch.

El switch recibe el número de arriba.

Para explicarlo de la forma más simple posible:

Cada botón tiene un código interno “12345”, ese código lo llamamos id(“botón1”).

Por así decirlo botón1==12345

A partir de aquí.

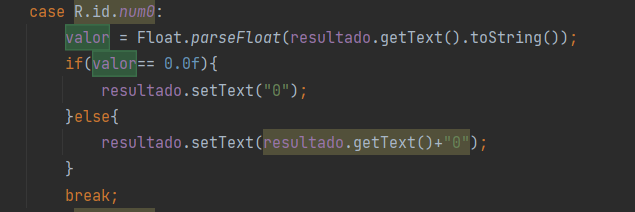
Como implementé el método setOnClick a mi OnCreate. El programa me obliga a tener un método OnClick

Para no hacerlo muy largo, creo un método void interno.

Ese método es el principal que recibe un número(“botón1==12345”) y con el switch digo:

Si he pulsado el “12345”(que es a su vez el botón 1) ejecuta esto:

Con esto explicado vamos con los casos de cada botón:



Para cada número es igual.

Valor es un float que acumula números.

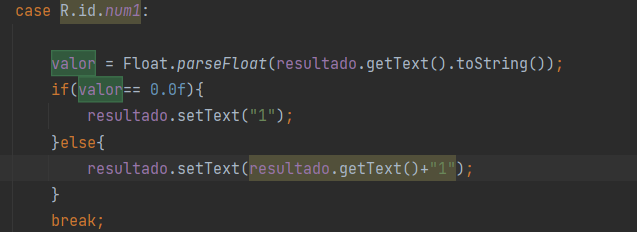
A valor le asigno lo que contiene el botón que acabo de pulsar.

Para ello paso lo que contiene la caja de texto(un text view) a String(cojo el resultado con getText y lo paso a String con .toString) y luego lo paso a un número.

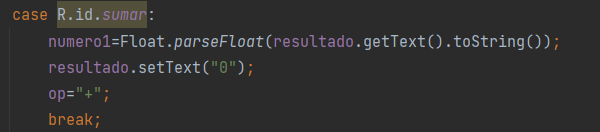
Si el valor númerico es 0 se deja en 0, y muestro 0.

Si no es 0, muestro lo que contenía antes(resultado.getText()) y concateno en este caso el 0 porque es el caso R.id.numero0

Vemos que el código es igual para todos los números.

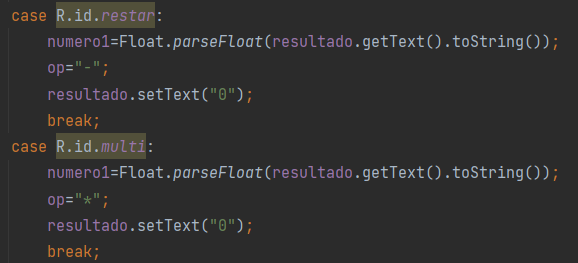


Si es 0 muestro el botón que acabo de pulsar y si es distinto a 0 concateno el contenido + el valor del botón.



Para la suma, cojo todo el contenido acumulado en la caja resultado y lo paso a número.

Pongo la caja a 0 y a la variable op(de operación) le asigno el valor “+” para utilizarlo más adelante.



Para la resta, multiplicación y división, el código es exactamente igual.

A la variable número1 le damos todo lo que contiene en la caja resultado y le asignamos a 0.



Para borrar, simplemente pongo el resultado a 0 y las variables numero1 y numero2 también se vuelven 0.



En el igual es donde se hacen los cálculos.

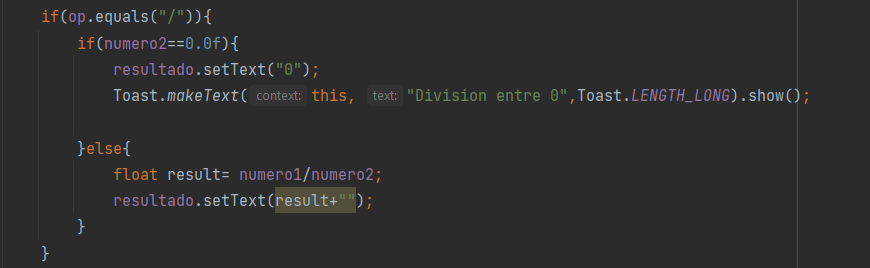
Lo primero, al presionar una operación, cojo todo lo que contiene resultado y lo meto dentro de la variable numero1.

Al pulsar igual, lo primero que hago es coger todo lo que contiene resultado (que se reseteó tras la operación) y lo meto dentro de la variable número2.

A partir de aquí es mucho más sencillo.

Cuando pulsamos cualquier operación le asignamos un valor a la variable op.

He utilizado ifs para comparar.



En este caso si op contiene / . Hago otro if para validar que el segundo número no sea 0.

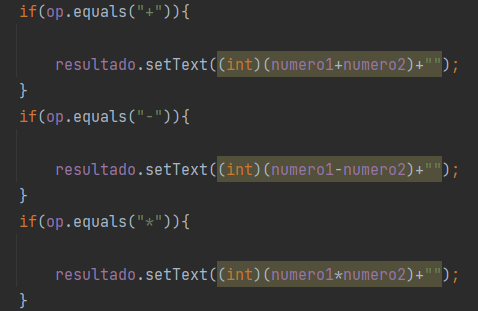
Si es 0, muestro el resultado en 0 y saco un mensaje por pantalla con el Toast.

En este caso mi mensaje es división entre 0.

Si el segundo número no es 0:

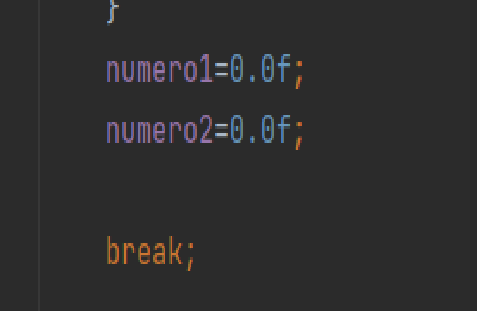
Creo una variable result y divido los dos números.

A continuación, muestro ese resultado por pantalla con el setText.

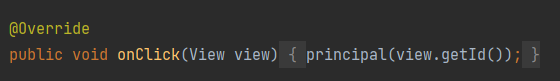


Dependiendo de lo que contenga op, sumará restará o multiplicará.

Por último, pongo las dos variables en 0.



Lo último sería llamarlo en el OnClick



Aquí es donde el programa coge un id con el getId , principal recibe ese Id y como 1234==boton1 ejecuta el código.