Instituto Politécnico Nacional

Centro de Investigación en Computación

Diseño e Implementación de Aplicaciones para Dispositivos Móviles

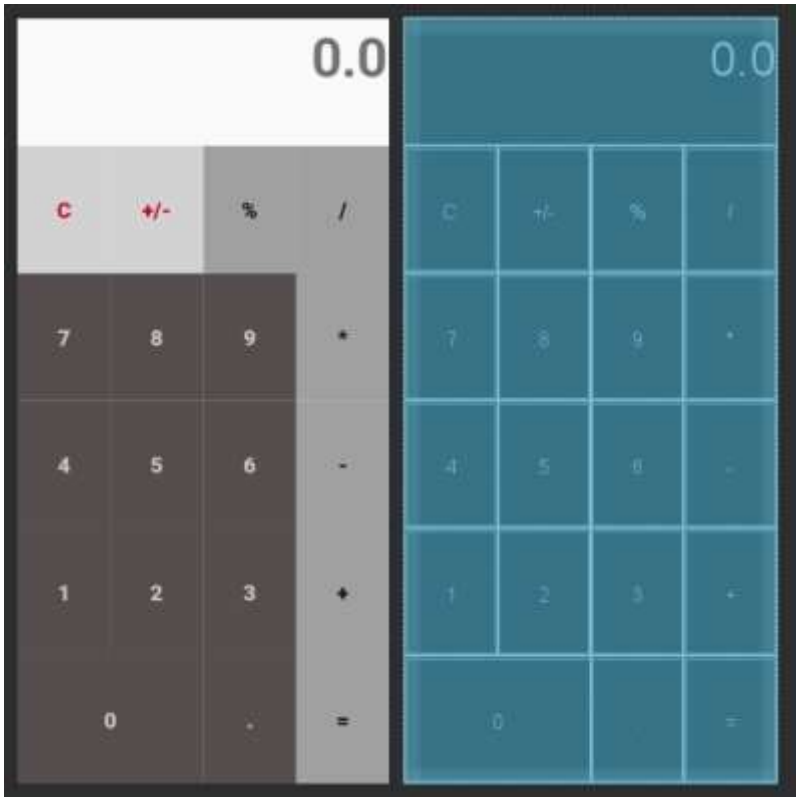
Dr. José Giovanni Guzmán Lugo

Práctica 1: Calculadora

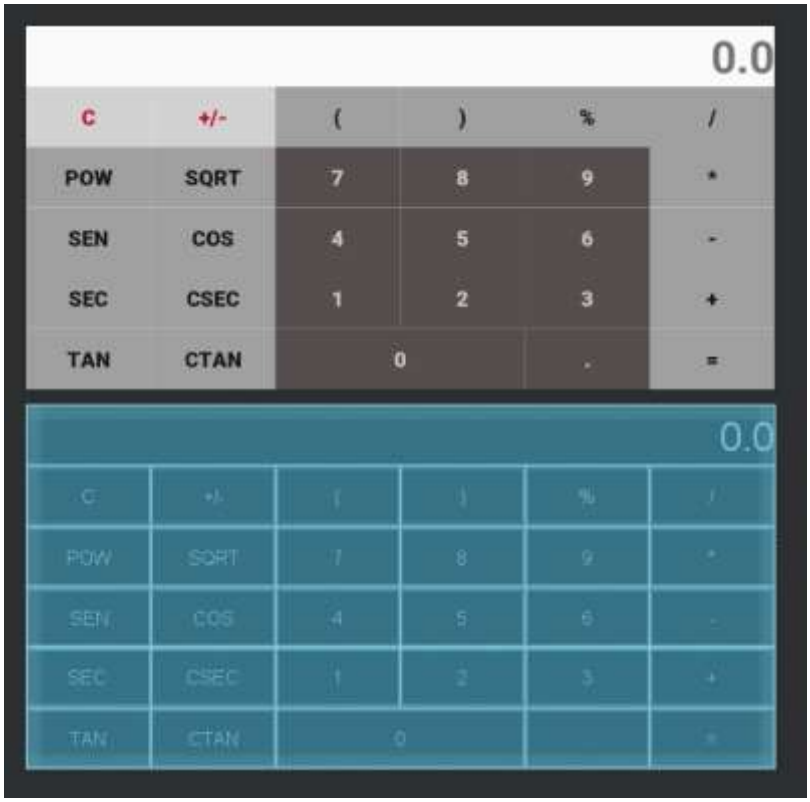
Gutiérrez Pacheco Julio Iván

26 de septiembre de 2019

Para realizar la Calculadora primero se realizo el diseño de la Interfaz Vertical con 19 botones y 1 TextView en 6 LinearLayaout con sus respectivos elementos en el strings.xml y colors.xml



Después de la Interfaz Horizontal con 29 botones y 1 TextView en 6 LinearLayaout con sus respectivos elementos en el strings.xml y colors.xml



Después utilizando el algoritmo visto en clase primero realizamos el Convertidor.java a la cual le puse 2 atributos un StringTokenizer y una pila, el StringTokenizer para la cadena a convertir y la pila para ir guardando los operadores.

Para el StringTokenizer utilice como separador el símbolo #.

Para el algoritmo utilice un while que va a sacar cada elemento del StringTokenizer, después dentro de un bloque de try-catch se verifica si el elemento es un número o un operador, si es un número se agrega a la cadena de salida, de lo contrario se mete en un switch case que evalúa lo que tiene hasta arriba en la pila (si no hay nada se inserta el operador) y se realiza una verificación de precedencia (tabla precedencia) si tiene mayor precedencia se mete en la pila de lo contrario se saca el elemento de hasta arriba se concatena a la cadena de salida y se ingresa el nuevo símbolo. En caso de paréntesis de cierre se sacan todos los elementos de la pila hasta encontrar el paréntesis de inicio.

Tabla de Precedencia

Salida\Entrada	+	-	*	/	^	raiz	%	Sen	Cos	Tan	Ctan	Sec	Csec	(
+	1	1	1	1	1	1	1	1	1	1	1	1	1	0
-	1	1	1	1	1	1	1	1	1	1	1	1	1	0
*	0	0	1	1	1	1	1	1	1	1	1	1	1	0
/	0	0	1	1	1	1	1	1	1	1	1	1	1	0
^	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Raíz	0	0	0	0	0	0	0	0	0	0	0	0	0	0
%	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Sen	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Cos	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Tan	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Ctan	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Sec	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Csec	0	0	0	0	0	0	0	0	0	0	0	0	0	0
(	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Después para el Evaluador.java le puse 2 atributos un StringTokenizer y una pila, el StringTokenizer para la cadena a evaluar y la pila para ir guardando los números.

Para el StringTokenizer utilice como separador el símbolo #

Para el algoritmo utilice un while que va a sacar cada elemento del StringTokenizer, después dentro de un bloque de try-catch se verifica si el elemento es un número o un operador, si es un número se agrega a la pila, de lo contrario se mete en un switch case que evalúa cual es el operador, en caso de ser binario se sustraen 2 elementos de la pila y se realiza la operación (utilizando el penúltimo, operador, último), en caso de ser unario solo se saca un elemento. En el caso del % se sacan dos elementos de la pila y se hace penúltimo * (último / 100)

Después se guarda el resultado de la operación en la pila.

Cuando se acaban los elementos el único elemento de la pila es el resultado.

Para los onClick cree 5 uno para los dígitos, otro para los operadores, otro para el cambio de signo, otro para el de limpiar la pantalla y otro para el igual.

El de los dígitos evalua que solo se pinte una vez el punto y se va guardando en una variable.

El de operadores reinicia el TextView, guarda el número obtenido en otra variable junto con el operador separados por un #, reinicia la variable del número y reinicia la bandera de punto.

El de cambio de signo revisa que la variable del número no tenga “-“ en el primer puesto de tenerlo lo quita, de lo contrario lo pone.

El de limpieza del TextView reinicia la variable del número y limpia el TextView.

El de igual termina la variable a evaluar y crea un objeto Covertidor y llama a su función, después crea un objeto Evaluador y llama a su función y pone el resultado en el TextView.