

1 Mejorando la calculadora

Materiales

- La calculadora realizada en clase, la cual se irá actualizando con nuevas funciones que deberás ir integrar a la práctica que entregues.

Requerimientos mínimos

1. Tu calculadora funciona con números enteros, es decir, si realizas $3/4$ el resultado será 0 en vez de 1.75. Realiza los cambios necesarios para que se muestren números de punto flotante.
2. Añade un botón a la UI para que el usuario pueda ingresar el punto decimal. Asegúrate de que solamente se puedan teclear números válidos (por ejemplo "192.168.2.20" no es un número válido).
3. Añade las siguientes operaciones de un solo operando:
 - +/-: cambia el signo del número en la pantalla.
 - sin: calcula el seno del número en la pantalla.
 - cos: calcula el coseno del número en la pantalla.
 - sqrt: calcula la raíz cuadrada del número en pantalla.
 - 1/x: invierte el número en la pantalla (ejemplo: 4 se vuelve 0.25).
 - 10^x : eleva 10 a la potencia x
4. Añade las siguientes operaciones de dos operandoos:
 - x^n : eleva el número x a la potencia n
 - root n : saca la raíz n del número en pantalla
5. Añade los siguientes botones de memoria:
 - Store: guarda el valor actual de la pantalla en la memoria. Este botón no debe modificar la pantalla.
 - Recall: devuelve lo almacenado en la pantalla.
 - Mem +: suma lo que la pantalla muestra a lo que está almacenado en la memoria. Este botón no debe modificar la pantalla.
 - Mem -: resta lo que está en la pantalla del contenido de la memoria.
 - MC: borra solamente el contenido de la memoria.
5. Añade un botón que borre todo (la pantalla, las operaciones pendientes y la memoria).
6. Añade un Label en algún lugar de la UI que muestre el estado de una operación de 2 operandos. Por ejemplo, si el usuario quiere hacer la operación " $3 + 4 * 5 =$ " este campo deberá comportarse de la siguiente manera:
 - No va a mostrar nada hasta que se presione "+" y entonces mostrará " $3 +$ "
 - Cuando se presione "4" se mostrará " $3 + 4$ "
 - Al presionarse " $*$ ", cambiará a " $7 *$ "
 - Al presionarse "5" se mostrará " $7 * 5$ "
 - Cuando se presione "=" se mostrará el resultado.
7. Cambia el ícono de la aplicación por uno apropiado a esta aplicación.

8. Implemente un botón de “backspace” para que el usuario pueda corregir algún error en lo que teclee. No es un botón de “deshacer”.
9. Mediante colores, “agrupa” los botones según su comportamiento. Ejemplo: un color para números, otros para ciertas operaciones, etc.
10. Añade un botón para π .

Tips

Estos son algunos consejos que pueden ser útiles en la elaboración de la tarea y son completamente opcionales.

1. El programa nunca debe tronar. Realiza las validaciones requeridas.
2. Para realizar la parte de la implementación del punto flotante, podrías utilizar algún método de la clase `String` de Kotlin. Consulta la documentación.
3. Ten cuidado en el caso en que el usuario inicie un nuevo operando usando el punto decimal (por ejemplo si quiere teclear “.5”). Atiende este caso de forma apropiada.
4. Consulta la documentación de Kotlin para la implementación de seno y coseno.
5. Las funciones de un solo operando (como raíz cuadrada o $1/x$) las puedes pensar como operaciones, de tal forma que no sea necesario escribir funciones específicas para ellas. Lo mismo aplica para los botones de operaciones de memoria. Crear métodos específicos para atender estos casos es un desperdicio de código. Estas funciones operan o asignan al operando actual como cualquier otra operación de un solo operando.
6. Para implementar el punto 6 de los requisitos mínimos, tendrás que modificar la API de `ModeloCalculadora` para compartir la información sobre si hay o no una operación de dos operandos pendiente de realizar

Puntos extra

Estas son algunas ideas que podrías implementar para ir obteniendo mayor experiencia.

1. Haz que tu calculadora sea más amigable con el usuario alertándolo cuando se realice una operación no permitida (como división entre cero o la raíz de un número negativo). Elige una de estas opciones:
 - puedes añadir otro `Label` en la UI que indique el error (+0.5 puntos), o
 - Usa un `Toast` (+2 puntos).
2. Añade un `Label` en la UI que muestre el contenido de la memoria. Esto requiere que `ModeloCalculadora` “exporte” esa información al controlador a través de su API (o sea, con un nuevo método en `ModeloCalculadora`) (+0.5 puntos)
3. Por defecto, las operaciones trigonométricas utilizan radianes. Implementa una forma para que el usuario elija entre operar en radianes o grados. Elige una de estas opciones:

- Puedes utilizar un `Button` y que su texto se modifique cada vez que el usuario cambie entre los modos (+0.5 puntos), o
- Una solución más elegante es utilizar un `Switch` que en cada extremo indique la opción en operación (lee la documentación) (+1 punto).
- Para cualquier caso, añade un `Label` en la UI que muestre “DEG” o “RAD” apropiadamente (+1 punto)

Evaluación

Cada requerimiento mínimo tienen el mismo valor, el cual se divide en los incisos que contenga.

Uno de los objetivos del semestre es que escribas código que compile sin errores ni warnings y que lo vayas mejorando a través de pruebas constantes.

Algunos motivos por lo que la calificación se afectará:

- El proyecto no compila (se evaluará con cero)
- El proyecto truena (se deducirá dos puntos y se evaluará lo que sea posible)
- El proyecto compila con warning (se deducirá una décima por cada uno)
- No se completaron los requerimientos mínimos
- Algún concepto fundamental no se respetó (por ejemplo división apropiada entre modelo, vista y controlador) (-1 punto)
- El código no está comentado. Los comentarios deben permitir la comprensión del código. Como mínimo se espera que se describa lo que cada método hace, las variables que recibe y lo que regresa. La cantidad de comentarios está en función de que tu código debe ser de fácil comprensión por alguien que lo lea (se asume que esa persona conoce el SDK). (-1 punto)
- El código está descuidado y es de difícil lectura (indentación inconsistente, escritura incorrecta de variables, objetos, clases y métodos, etc) (-1 punto)
- Por cada error de ortografía se deducirá una décima.

Archivos a entregar

Las tareas y proyectos se deben entregar en un .zip que contenga el subdirectorio de tu proyecto. Asegúrate de enviar todos los archivos necesarios para la compilación. El archivo .zip se debe llamar como tu número de cuenta y lo deberás subir a Brightspace en el espacio correspondiente a la actividad. Puedes realizar varias entregas, pero sólo se conservará y evaluará la última. No se permitirá entregas después de la fecha límite