

Algoritmos y Patrones de Almacenamiento No Lineales Orientados a Objetos

Primer Proyecto

Prof. Miguel A. Pizaña

9 de febrero de 2015

1. **Objetivo:** Construir una calculadora en java.
2. **Apariencia:** La calculadora tendrá una pantalla y 20 botones acomodados de la siguiente manera:

pantalla			
()	C	±
1	2	3	+
4	5	6	-
7	8	9	×
.	0	=	/

3. **Funcionamiento:** Al iniciar, la pantalla debe mostrar el valor “0.0”. El usuario escribirá expresiones aritméticas con el ratón haciendo click sobre los botones de la calculadora o usando directamente las teclas de su teclado.

Aparte de las asociaciones evidentes del teclado (números, operaciones, paréntesis), se tendrán las siguientes asociaciones: La operación *borrar todo* (o “resetea”), estará asociada al botón “C” y a las teclas **C**, **Supr.** y **Back_Space**. La operación ‘=’ estará asociada a las teclas “=” y a ambos **Enter**’s. Entiéndase que el botón y la tecla “-” solamente se usan para la operación binaria “resta” y no se usa para indicar el signo de un operando, es decir, no es una operación unaria. La operación unaria correspondiente se realiza por medio del botón ± o por medio de la tecla ‘s’.

Recuerde que los operadores “-” y “/” son asociativos a la izquierda, en particular, el resultado de “9-4-2=” es 3 y no 7.

Cuando el usuario esté escribiendo un número, la pantalla de la calculadora deberá mostrar la parte del número que ya se haya escrito hasta el momento. El resto del tiempo la pantalla deberá mostrar el resultado parcial de todas las operaciones previas que ya puedan realizarse. En ningún momento deben aparecer los operadores en la pantalla ni ninguna otra cosa que no sea un número: un signo (sólo si el número es negativo), dígitos (al menos uno), punto decimal y dígitos de nuevo (al menos uno). **Única excepción:** Si las operaciones indicadas por el usuario involucran una división entre cero, la pantalla deberá presentar en pantalla la leyenda “Error” y resetear todo el sistema interno para estar listo para una nueva operación.

La calculadora respetará la precedencia de los operadores de modo que si el usuario escribe “1+2×3=” el resultado será 7 y no 9. A continuación ponemos algunos ejemplos para aclarar este funcionamiento. En el primer renglón de cada tabla aparecen de izquierda a derecha los botones o teclas que ha ido oprimiendo el usuario. En el segundo renglón, aparece lo que la calculadora debe mostrar en su pantalla inmediatamente después de cada botón o tecla oprimido.

1	+	2	×	3	=
1.0	1.0	2.0	2.0	3.0	7.0

2	×	3	-	5	=
2.0	2.0	3.0	6.0	5.0	1.0

2	3	+	1	3	-	3	5	+	1	2	=
2.0	23.0	23.0	1.0	13.0	36.0	3.0	35.0	1.0	1.0	12.0	13.0

1	2	.	5	3	+	4	.	7	/	2	.	5	+	6	=
1.0	12.0	12.0	12.5	12.53	12.53	4.0	4.0	4.7	4.7	2.0	2.0	2.5	14.41	6.0	20.41

1	+	2	×	3	/	5	+	6	=	+	3	=	×	5	=
1.0	1.0	2.0	2.0	3.0	6.0	5.0	2.2	6.0	8.2	8.2	3.0	11.2	11.2	5.0	56.0

2	×	3	-	4	×	5	+	2	=	/	6	/	2	+	1	=
2.0	2.0	3.0	6.0	4.0	4.0	5.0	-14.0	2.0	-12.0	-12.0	6.0	-2.0	2.0	-1.0	1.0	0.0

4. **Errores:** Los errores de usuario (al escribir “3.4.5”, “3++2” o “3+4w” por ejemplo) deberán ser indicados cambiando el color de la pantalla momentáneamente. El botón o tecla que haya causado el error, deberá ser ignorado y la calculadora deberá continuar funcionando normalmente. En ningún caso, debe reportarse el error con un mensaje al usuario, ni abortarse o interrumpirse la operación en curso.

No deberían ocurrir errores en su programa, pero debe haber código que maneje todos los posibles errores internos (estado desconocido, caracter desconocido, pila vacía, etc). Deberán tener los siguientes métodos en la clase Evaluador: `fatal(String msg)`, `error(String msg)`, `warning(String msg)`, `log(String msg)`, para reportar de manera uniforme los diferentes tipos de errores y la bitácora.

Tanto los errores de usuario como los errores internos deben ser reportados con mensajes a la consola (`System.out.println()`). También, la calculadora reportará por la consola una bitácora de eventos indicando cada cosa que haga.

5. **Entrega:** Los proyectos se entregarán por email de manera individual, a más tardar el domingo 1 de Marzo a las 24hrs. Se penalizará la entrega tardía (10 puntos luego de la hora límite, 10 puntos adicionales de penalización por cada 48 horas adicionales de retraso hasta un máximo de 30 puntos de penalización).

Deberán entregar un jar ejecutable, que se llame: <tu número de cuenta>.jar El jar ejecutable debe contener tanto el código en java como las clases necesarias para su ejecución. El jar ejecutable deberá correr haciendo click sobre él en Linux, Windows y Mac. Deberán enviar el archivo a la siguiente cuenta: mpizana@gmail.com. El asunto del email (el “subject”) debe ser “Calculadora” (sin las comillas).

Si hay proyectos que sean copias unos de otros, serán penalizados fuertemente, por lo que se recomienda que desarrollen su proyecto de manera completamente independiente.

6. **Evaluación:** La evaluación se realizará de acuerdo con tabla anexa. Empiezan con una puntuación inicial de 100 puntos y a partir de ahí se pueden sumar o restar puntos de acuerdo con los criterios establecidos. En cada rubro, se encuentra entre paréntesis el puntaje máximo que puede ser restado (o sumado) a su calificación.

Algoritmos y Patrones de Almacenamiento
No Lineales Orientados a Objetos
Primer Proyecto
 Prof. Miguel A. Pizaña
 24 de febrero de 2014

Nombre del alumno:

Número de cuenta:

Puntuación inicial.		100
No funciona o no es un jar ejecutable o el jar no contiene el código fuente o no se entrega en la cuenta de email indicada o el archivo no tiene el nombre que debe o el email no tiene el asunto "Calculadora".	(-100)	
El código del programa es demasiado parecido a otro programa entregado.	(-200)	
El alumno entregó más de una versión de su programa o más de un archivo.	(-20)	
La calculadora tiene menos o más de 20 botones o los botones no son los que deben o el acomodo de los botones no es el indicado.	(-20)	
La pantalla de la calculadora no muestra el valor 0.0 al iniciar.	(-10)	
Los operadores "-" o "/" no asocian a la izquierda.	(-50)	
La calculadora no respeta la precedencia de los operadores.	(-50)	
El programa no es estable.	(-40)	
El código del programa no es claro o no hace uso de la "programación defensiva".	(-20)	
El programa da resultados aritméticos erróneos.	(-40)	
El programa no muestra en pantalla los resultados parciales que debe.	(-40)	
El programa presenta en algún momento algo distinto de un número en su pantalla (excepto "Error" en caso de división entre cero).	(-40)	
El programa no cambia momentáneamente el color de la pantalla ante errores de usuario.	(-20)	
El programa no reporta errores o no reporta la bitácora por medio de la consola.	(-30)	
El programa reporta errores al usuario de manera distinta a lo indicado en los tres puntos anteriores.	(-30)	
El programa no ignora botones y teclas que representen errores de usuario.	(-20)	
El programa aborta, o interrumpe el cálculo actual ante errores de usuario.	(-30)	
El programa se entrega tardíamente.	(-30)	
El programa maneja correctamente errores de desbordamiento.	(+20)	
El alumno se esmeró en darle a su programa una apariencia agradable (colores adecuados; separación, forma, o estilo de botones, tamaños y tipos de letra, etc).	(+20)	
Puntuación total:		