



PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
ESCUELA DE INGENIERÍA
DEPARTAMENTO DE CIENCIA DE LA COMPUTACIÓN

IIC2233 - Programación Avanzada
1º semestre 2015

Tarea 4

Calculadora Gráfica

1. Objetivos

- Implementación de una interfaz gráfica.
- Manejo de errores.

2. Problema

Para esta tarea deberá implementar una interfaz gráfica que permita al usuario realizar operaciones aritméticas usando bloques.

3. Especificaciones del sistema

Bloques: Los bloques son la base del programa. Estos reciben entradas y entregan resultados de salida. Pueden ser conectados en serie para realizar operaciones complejas (como $\sin(2) + \ln(3)^4$).

El sistema debe soportar las siguientes operaciones:

- Input: No recibe entradas. Se le debe poder asignar un valor que entregue como input a los otros bloques
- Suma: Recibe 2 entradas y entrega su suma.
- Resta: Recibe 2 entradas y entrega su resta.
- Multiplicación: Recibe 2 entradas y entrega su multiplicación.
- División: Recibe 2 entradas (a y b) y entrega su división (a/b).
- Potencia: Recibe 2 entradas (x y n) y entrega su potencia (x^n).
- Valor Absoluto: Recibe una entrada y entrega su valor absoluto.
- Logaritmo Natural: Recibe una entrada y entrega su logaritmo natural.
- Seno: Recibe una entrada y entrega el seno (se evalúa en radianes).

- Coseno: Recibe una entrada y entrega el coseno (se evalúa en radianes).
- Tangente: Recibe una entrada y entrega la tangente (se evalúa en radianes).
- Mínimo: Recibe n entradas y entrega el mínimo entre ellas.
- Máximo: Recibe n entradas y entrega el máximo entre ellas.
- Output: Recibe una entrada y la muestra en pantalla.

El sistema debe permitir usar números trascendentes sin tener que escribir una aproximación decimal de ellos, es decir, el usuario debe poder escribir π y e , por ejemplo, para poder usar sus valores.

La salida de un bloque puede conectarse a cualquier número de entradas de otros bloques, pero la entrada de un bloque puede ir conectada a una sola salida.

En caso de dejarse una entrada de un bloque desconectada, debe avisarse al usuario. Esto no es necesario en caso que se deje una salida desconectada.

4. Interfaz Usuaría

La interfaz consiste de una área de trabajo donde el usuario puede agregar nuevos bloques y conectarlos entre sí.

El usuario debe poder agregar la cantidad de bloques que quiera al área de trabajo. Cómo se agregan los bloques al área de trabajo queda a criterio suyo. Los bloques agregados deben poder arrastrarse alrededor del área de trabajo, y mantener sus conexiones al hacerlo. Debe darse la opción de borrar bloques de manera fácil.

Cómo se realizan y presentan las conexiones queda a criterio suyo, mientras sea fácil de usar y claro. Debe darse la opción de borrar conexiones de manera fácil.

El programa no debe caerse bajo ninguna condición, si el usuario hace una operación que no se puede manejar se debe avisar en la misma interfaz.

Es de suma importancia que la interfaz usuaria sea clara y atractiva.

A continuación se presenta un ejemplo de una implementación de la expresión $-128 + -1$ usando bloques:

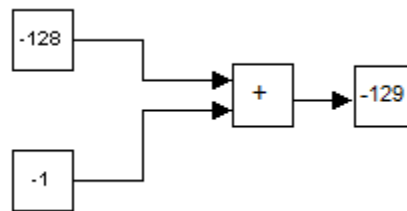


Figura 1: Implementación de $-128 + -1$

A continuación se presenta un ejemplo de una implementación de la expresión $\sin(2) + (\ln(3))^4$ usando bloques:

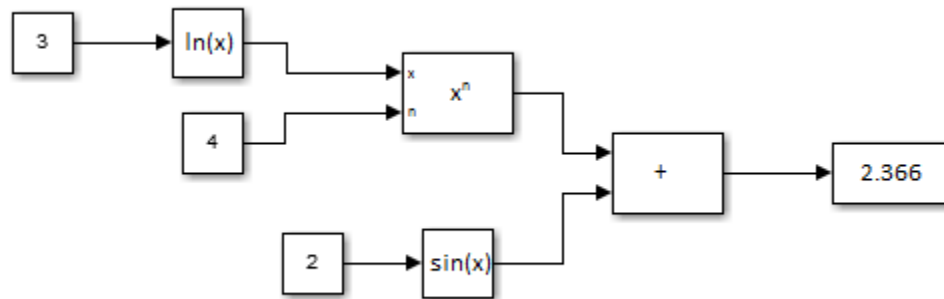


Figura 2: Implementación de $\sin(2) + (\ln(3))^4$

5. Restricciones y alcances

- Tu programa debe ser desarrollado en Python 3.4
- Todas las funcionalidades deben ser testeadas, esto incluye comprobar que las expresiones estén bien calculadas.
- Su código debe seguir la guía de estilos PEP8
- Si no se encuentra especificado en el enunciado, asuma que el uso de cualquier librería Python está prohibido. Pregunte por foro si se pueden usar librerías específicas.
- Todo error debe ser controlado, para que el programa no se caiga.
- Debe usar la librería **PyQt4** para realizar la tarea.
- Puede usar la librería **NumPy** para realizar la tarea.
- El ayudante puede castigar el puntaje¹ de tu tarea, si le parece adecuado. Se recomienda ordenar el código y ser lo más claro y eficiente posible en la creación algoritmos.
- Debe adjuntar un archivo **README.md** donde comente sus alcances y el funcionamiento de su sistema (*i.e.* manual de usuario) de forma *concisa* y *clara*.
- Cree un módulo para cada conjunto de clases. Divídalas por las relaciones y los tipos que poseen en común.
- Cualquier aspecto no especificado queda a su criterio, siempre que no pase por encima de ningún otro.

6. Entrega

- **Fecha/hora:** 22 de Mayo de 2015 / 23:59.
- **Lugar:** GIT - Carpeta: Tareas/Tarea_04

Tareas que no cumplan con las restricciones señaladas en este enunciado tendrán la calificación mínima (1.0).

¹Hasta -5 décimas.