**Laboratorio Unidades 1 y 2**

En las unidades 1 y 2 de nuestro curso hemos trabajado sobre el concepto de algoritmo y su aplicación en la solución de problemas. Además, el análisis de problemas a través de diagramas de flujo y su posterior traducción a un lenguaje de programación, para lo cual, usamos tipos de datos primitivos y cadenas de texto, procesando estos datos con estructuras de control condicionales e iterativas. Adicionalmente, hemos trabajado sobre la división del problema en problemas más pequeños cuya solución es más sencilla de traducir al lenguaje de programación usando métodos auxiliares. Hemos visto la importancia de la documentación a través de contratos y las buenas prácticas al programar que le brindan seguridad a nuestras aplicaciones. El presente laboratorio les presenta una actividad en la cual se requiere aplicar todos los conocimientos adquiridos en estas unidades y verificar de esta manera el cumplimiento de los objetivos que han sido planteados para las unidades 1 y 2 descritos en el programa del curso. Para llevar a cabo este ejercicio es necesario realizar las actividades listadas a continuación:

**Actividades**

Lleve a cabo las siguientes actividades de cada una de las etapas de desarrollo de software:

1. Análisis del problema (Definición de uno o más diagramas de flujo que representen la solución planteada)
2. Implementación en Java. Incluya en la implementación, los comentarios descriptivos sobre los atributos y métodos de cada clase. Recuerde que todos los artefactos generados de fase de diseño e implementación deben ser en inglés.
3. Documentación en JavaDoc (Debe entregarse el JavaDoc generado).
4. Usar GitHub como repositorio de código fuente utilizando la estructura de carpetas aprendida en clase. Recuerde que se debe evidenciar su avance a través de los días en el laboratorio.

Recuerde que puede encontrar la Rúbrica laboratorio en el siguiente enlace.

**Nota:** Usted debe entregar un archivo en formato pdf con toda la documentación (análisis) y la URL de su repositorio git Hub donde se deben encontrar los archivos de codificación en sus respectivos paquetes.

Tenga en cuenta que su repositorio gitHub debe presentar una estructura base como por ejemplo:

Calculator/

src/ bin/ docs/

Dentro de los directorios src/ y bin/ estarán presentes estos directorios, representando cada uno de sus paquetes:

ui/

El directorio src (source code) contiene sus clases .java dentro del directorio ui (por ahora). Por otro lado el directorio bin (binary files) contiene los archivos .class en el directorio ui.

Su código debería compilar de acuerdo con lo explicado en la diapositiva 13 de esta presentación: <http://tinyurl.com/y3bd9bg2>

A continuación, encontrará un enunciado que narra de forma detallada la situación problemática que se espera usted solucione.

**Enunciado**

En los cursos de matemáticas de la universidad los estudiantes usualmente usan una calculadora cientifica como elemento de apoyo. En ocasiones los estudiantes olvidan esta importante herramienta de apoyo por lo cual le solicitan ayuda para desarrollar una aplicación que asista a los estudiantes en caso de olvidar su calculadora. En la calculadora construida por usted, se requiere que ofrezca las siguientes funcionalidades sobre números reales:

1. Suma.
2. Resta.
3. División.
4. Multiplicación.
5. Módulo.
6. Seno.
7. Coseno.
8. Tangente.
9. Logaritmo base 10 y base n.
10. Raíz cuadrada.
11. Raíz n-ésima.
12. Potenciación.
13. Base 10 de un número x.
14. Memoria de los resultados de las últimas 10 operaciones. (Estos resultados pueden ser usados en la operación actual del usuario).
15. El factorial de un número.
16. La conversión de grados a radianes y viceversa.
17. La conversión entre binarios, decimales y hexadecimales.

Su calculadora debe soportar los números del 0 al 9, las letras de la A a la F, el número Pi.

Las operaciones de la 1 a la 10 deben ser implementadas por usted sin el uso de librerías externas, para el resto de operaciones usted puede consultar el API de Java para facilitar su implementación.

Su calculadora debe soportar resolver ecuaciones en 2 modos, el primer modo es una operación a la vez, y la segunda opción es un flujo de operaciones.

Un ejemplo de la opción 1 podría ser:

8

+

2

||

10

¿Desea continuar con otra operación matemática?

Un ejemplo de la opción 2 podría ser:

8

+

2

10 // Lo imprime el programa

-

3

7 // Lo imprime el programa

\*

3

21 // Lo imprime el programa

# // Fin del modo 2 y regreso al modo 1