

# Proyecto 1

## Estructuras de Datos

### Enero – mayo 2019

Hacer una calculadora que pueda realizar operaciones aritméticas sobre números reales, tanto negativos como positivos, y que cuente con una interfaz gráfica amigable para el usuario. Las operaciones a realizar las indica el usuario a través de expresiones en notación infija que pueden utilizar los operadores +, -, \* y /, y además pueden tener paréntesis para alterar el orden estándar de ejecución de los operadores. La interfaz necesita contar con botones para las cuatro operaciones mencionadas anteriormente y los dos paréntesis (izquierdo y derecho), también debe tener un botón para cada uno de los diez dígitos 0-9, uno para el punto decimal, uno para indicar “cambio de signo” (para poder expresar valores negativos), y uno con el símbolo = para que el programa comience a evaluar la expresión que el usuario haya indicado/ingresado. Puede ser conveniente que la calculadora también tenga un botón para “limpiar el contenido” de la ventana de despliegue (y así pueda el usuario empezar a indicar una nueva expresión a evaluar).

Cuando el usuario apriete el botón = se deben realizar tres pasadas sobre la expresión que ingresó:

- 1) Verificar su validez (por ejemplo, asegurarse de que los paréntesis estén balanceados, asegurarse de que no se hayan puesto dos operadores seguidos como “1+\*53.2”, y filtrar otros posibles errores sintácticos parecidos a éste).
- 2) Convertir la expresión (que ahora se sabe que es sintácticamente válida, y que está en notación infija) a su equivalente en notación postfija.
- 3) Evaluar la expresión en notación postfija para conocer su resultado global. El resultado debe mostrarse en la ventana de despliegue de la calculadora (la cual también debe utilizarse para irle mostrando al usuario los botones que ha ido apretando antes de apretar el botón = y para mostrar cualquier mensaje de error que pueda ser necesario desplegar).

#### **Especificaciones extras:**

- Para desarrollar el proyecto los alumnos deben formar equipos de trabajo (4 integrantes).
- Los alumnos deben leer el documento “Cultura de trabajo en equipo” (archivo *TrabajoEnEquipo2017.doc*, que está en Comunidad).
- Cada alumno debe identificar su estilo como parte de un equipo de trabajo (usar documento “*The team-player survey*”, archivo del mismo nombre que está en Comunidad).
- Se sugiere que los equipos estén formados por un miembro de cada uno de los estilos (*contribuidor, colaborador, comunicador, crítico*), si es posible.
- A más tardar el día **12/02/2019** el equipo deberá enviar al profesor ([guardati@itam.mx](mailto:guardati@itam.mx)) la siguiente información:

- 1) Integrantes del equipo
- 2) Objetivo(s) del equipo
- 3) Plan de tareas/actividades: lista de las tareas identificadas que deben llevarse a cabo para alcanzar los objetivos
- 4) Fechas límites: fecha límite en la que deberá estar concluida cada tarea para poder alcanzar los objetivos en tiempo y forma
- 5) Un párrafo explicando cómo se organizaron para trabajar colaborativamente y asegurarse de incluir el esfuerzo de todos los integrantes del equipo

**NOTA:** los puntos 3 y 4 pueden entregarse juntos

**Fecha de entrega del proyecto: 21 de febrero de 2019**

**Entregables:**

- El **proyecto de Netbeans completo** comprimido enviarlo por correo electrónico a: [guardati@itam.mx](mailto:guardati@itam.mx) Si se entrega el proyecto incompleto o las clases por separado, se considerará NO entregado.
- El código debe estar debidamente documentado usando Javadoc.
- El código debe ser probado usando JUnit.
- Se debe entregar un reporte escrito (impreso) sobre el proyecto. Ver lineamientos e información adicional publicados en Comunidad.
- El reporte debe tener un apéndice con el código (debidamente documentado) del programa. En el apéndice NO se debe incluir el código correspondiente a la interface PilaADT y a la clase PilaA (hechas en clase).
- Rúbrica del SO5 (está en Comunidad, carpeta Trabajo en equipo) llenada por cada uno de los integrantes del equipo.