



### Hoja de trabajo No. 3

**Realizar:** Programa para evaluar expresiones postfix.

**Realizarse:** en forma **INDIVIDUAL**.

**Fecha de entrega:** viernes 20 de agosto

**Objetivos:**

- Utilización de genéricos.
- Diseño del ADT para pilas (stack).
- Implementación de pilas con una lista simplemente encadenada.
- Control de versiones del programa.
- Emplear JUnit para casos de prueba.

**Programa a realizar:**

Su programa debe leer, de la manera que mejor le convenga a usted, una expresión en formato Postfix y producir el resultado de la misma.

Por ejemplo, el cálculo de la expresión postfix:  $1\ 2\ +\ 4\ *\ 3\ +$

La expresión es evaluada de izquierda a derecha utilizando una pila: el resultado es 15

**NOTA:** Observe que es la misma tarea de la hoja número 2, pero la pila es implementada usando una lista simplemente encadenada.

**Tareas:**

- Diagrama UML de clases. NO utilice ingeniería reversa. Muestre la relación entre las clases que usa para la pila y las utilizadas para la lista simplemente encadenada.
- Construir la interfaz de la Pila, su clase abstracta y la clase de implementación con una lista simplemente encadenada. Debe utilizar genéricos.
- Su programa principal debe permitir cambiar la clase pila, sin que se vea afectado en su funcionamiento.  
Nota: en su control de versiones se debe notar que el único cambio realizado en el código de su programa principal es el cambio de la clase pila para que utilice la nueva implementación de listas simplemente encadenadas.
- Debe dejar evidencia de todo el desarrollo en el repositorio de Subversion o sistema similar para control de versiones. Indicar como acceder a su repositorio y si es necesario, agregar a su catedrático y auxiliar para que tengan acceso al mismo.
- Incluya pruebas unitarias con JUnit para la clase pila.

Debe subir a Sakai todos los productos elaborados en los incisos a, b, c, e y los enlaces a su repositorio de Subversion.



**Calificación:** deben existir los diagramas de Clases para que sea calificado su programa.

Aspecto	Puntos
Estilo de codificación: comentarios, indentación, nombres de variables significativas.	5
Documentación generada con Javadoc, tiene precondiciones y postcondiciones en los métodos del ADT	5
Uso del repositorio: existen más de tres versiones guardadas, la última versión es igual a la colocada en el SAKAI	10
Diagrama de clases: muestran la abstracción y encapsulación de las operaciones.	5
Implementación de la lista simplemente encadenada con genéricos	10
ADT pila, con uso de genéricos.	30
Funcionamiento del programa	25
Pruebas JUnit para las operaciones de la pila.	10
<b>TOTAL:</b>	<b>100</b>