

Hoja de trabajo No. 2

Realizar: Programa de implementación de pilas.

Realizarse: en forma **INDIVIDUAL**.

Fecha de entrega: miércoles 22 de julio

Objetivos:

- Definición de las operaciones de una pila genérica.
- Utilizar dos implementaciones posibles para la matriz genérica: arreglos y listas.
- Implementar las operaciones de listas encadenadas (Singly linked lists)
- Utilizar Subversion para guardar las versiones del programa.
- Utilizar JUnit para los casos de prueba del programa.

Programa a realizar:

Se debe implementar el mismo programa de manejo de un parqueo, elaborado en la Hoja de Trabajo no. 1.

El programa deberá poder utilizar dos implementaciones distintas de la pila, cambiando solamente las declaraciones de los objetos que sean instanciados, por las clases que contienen la implementación. Por ejemplo si se tiene una pila que representa un espacio de estacionamiento para los diez vehículos, entonces puede hacer esto¹:

```
Stack <placa> miEspacioEstacionamiento = new ImplementacionUnoStack <placa>();
```

Y luego el auxiliar deberá poder sustituir esa línea por:

```
Stack <placa> miEspacioEstacionamiento = new ImplementacionDosStack <placa> ();
```

Y su programa debería funcionar EXACTAMENTE igual usando cualquiera de sus implementaciones de la pila.

Se debe por lo tanto:

- Construir una interfaz genérica para la pila.
- Construir una clase genérica que implemente esa interfaz utilizando arreglos.
- Construir una interfaz genérica para la lista. (No es necesario definir todos los métodos de la lista, use solo aquellos que serán útiles en su implementación del stack).
- Construir una clase genérica que implemente la interfaz de lista usando listas encadenadas simples.

OPCIONALMENTE: puede construir una clase abstracta que implemente la interfaz de lista, y luego una clase que extienda esa definición con la implementación de la lista encadenada. Esto será requerido para la hoja de trabajo #3.

Es obligatorio que todas las definiciones de interfaces y clases usen genéricos.

Se iniciará a utilizar la sentencia **assert** para verificar en tiempo de corrida que se cumplan las condiciones que necesita el programa para funcionar correctamente.² Puede utilizarla en algunos de los métodos de la implementación de la lista encadenada.

Tareas:

- Construir el programa que simule el parqueo, con pilas implementadas por el programador.
- Debe desarrollar la implementación de la lista encadenada.
- Debe desarrollar la implementación de la pila usando arreglos.
- Debe tenerse dos implementaciones para las pilas: con arreglos y con listas encadenadas.
- Debe dejar evidencia de todo el desarrollo en el repositorio de Subversion.

¹ Note la simplificación, ya que no se declaró en este ejemplo un arreglo de stacks.

² El libro muestra algunos ejemplos del uso de assert en la implementación de la lista encadenada, pero usa una clase desarrollada por el autor y no la oficial de Java. Por ello debe verificar que se use la sintaxis Java.

- f. Debe implementar casos de prueba, como mínimo para cada operación de la lista encadenada. Estos casos de prueba también deben estar en el repositorio de Subversion.
- g. El programa debe tener excepciones. Cada implementación también debe definir sus excepciones.
- h. Elaborar un video con la operación de su programa. Colóquelo en un sitio externo al Sakai y solo mande su enlace.

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

Calificación: su programa debe funcionar para ser calificado.

| Aspecto | Puntos |
|--|------------|
| Estilo de codificación: comentarios, indentación, nombres de variables significativas. Uso de excepciones. | 5 |
| Documentación generada con Javadoc | 5 |
| Utilización de assert en por lo menos un método de la lista encadenada. | 5 |
| Implementación de la pila genérica con arreglos | 5 |
| Implementación de la lista encadenada genérica | 20 |
| Implementación de la pila genérica con listas encadenadas. | 5 |
| Casos de prueba: debe existir como mínimo uno por cada operación de la lista encadenada implementado en JUnit. | 10 |
| Uso del repositorio: existen más de tres versiones guardadas, la última versión es igual a la colocada en el SAKAI | 10 |
| Video que muestre el funcionamiento completo del programa (casos usuales, protección de errores de usuario) | 10 |
| Funcionamiento del programa del parqueo | 25 |
| TOTAL: | 100 |