Clase 2 - 3 – Implementación estática

Ejercicio 1

Codificar la interfase del TDA PilaAlt2 cuya definición está comentada.

```
/**@author
 * @Pila es una estructura que permite almacenar una colección de valores, eliminarlos y recuperarlos, con
 * a particularidad de que el elemento que se recupera o elimina es el último que ingresó.
 * @Tarea Apilar agrega un elemento. Precondición la pila debe estar inicializada
 * @Tarea TopeSacar devuelve el último elemento agregado a la pila y lo elimina el último elemento.
 * Precondición la pila debe estar inicializada y no vacía.
 * @Tarea pilaVacia indica si la pila contiene elementos o no. Precondición la pila debe estar inicializada.
 * @Tarea InicializarPila inicializa la estructura de la pila
*/
public interface PilaAlt2TDA {
}
```

Ejercicio 2

Codificar la implementación tope al inicio para la interfase del TDA PilaAlt2.

```
/**@author
 * @Implementación tope al inicio */
public class PilaAlt2 implements PilaAlt2TDA {
}
```

Ejercicio 3

Completar la implementación estática de una Cola con almacenamiento circular y probarla.

Ejercicio 4

Codificar la implementación primero al inicio y probarla

Ejercicio 5

Completar la implementación estática de una Cola con prioridad.

Ejercicio 6

En que se modificaría la implementación del ejercicio 5 si en lugar de ordenar las prioridades en forma ascendente se ordenan en forma descendente.

Ejercicio 7

Podría en lugar de usarse un vector de Elem usado en el ejercicio ejemplo utilizarse dos vectores, uno almacenaría las prioridades y el otro los valores. Como se codificaría en la clase colaPrioridad2 las variables de clase y los métodos: inicializarCola, acolarPrioridad, prioridad y primero.

Ejercicio 8

Codificar los siguientes métodos y probar los utilizando las implementaciones de los TDA de enteros definidos en la primera clase.

1. Pase los valores de un conjunto a otro
2. Copie una cola con prioridad
3. Invierta una cola sin usar pilas y/o colas con prioridad.
4. Ordene en forma ascendente los valores de una cola
5. Verifique si los elementos de dos conjuntos son iguales.
6. Dada una cola con prioridad genere otra cola con prioridad de tal manera que no se repitan las prioridades. En caso de que haya varios valores con igual prioridad se le asociará el promedio de los valores que tenía asociada esa prioridad.
7. Dada una cola con prioridad genere otra cola con prioridad de tal manera que no se repitan los valores. En caso de que haya valores iguales con distinta prioridad, la prioridad asociada a ese valor será el promedio de las prioridades que tenía asociado ese valor.
8. Dado dos conjuntos generar un tercer conjunto resultado de realizar la unión entre ellos.
9. Dado dos conjuntos generar un tercer conjunto resultado de realizar la intersección entre ellos.
10. Dado dos conjuntos generar un tercer conjunto resultado de realizar la diferencia entre ellos.
11. Dado dos conjuntos generar un tercer conjunto resultado de realizar la diferencia simétrica entre ellos.
12. Devuelva el resultado de verificar si un conjunto está incluido en otro.