

TRABAJO PRÁCTICO N ° 4

Listas e Iteradores

Departamento de Ciencias e Ingeniería de la Computación - U.N.S.

Ejercicio 1:

Con el objetivo de implementar el TDALista con una estructura doblemente enlazada con centinelas:

- Utilice la interface PositionList<E> dada por la cátedra;
- Implemente la clase DNodo<E> que implemente la interface Position<E>;
- Implemente las excepciones pertinentes;
- Implemente un iterador para la lista programada.
- Ejecute el tester JUnit provisto por la cátedra para verificar la correctitud de su TDALista.

Ejercicio 2:

Agregue un método a la lista programada en el ejercicio 1 tal que reciba dos elementos, *e1* y *e2*, y modifique la lista receptora del mensaje de la siguiente manera:

- Deberá agregar a *e1* como segundo elemento de la lista;
- Deberá agregar a *e2* como ante-último elemento de la lista;

Considera casos especiales: ¿Qué sucede si la lista está vacía?, ¿Qué sucede si la lista tiene un elemento?. Recuerde que tiene total acceso a la estructura.

Ejercicio 3:

Dada una PositionList<E> *l* y un elemento genérico *e1*, escriba un método tal que determine si *e1* se encuentra en la lista *l*. Para resolver este ejercicio deberá hacer uso del iterador provisto por el TDALista. Compare los elementos por equivalencia.

Ejercicio 4:

Escriba un método tal que dada una PositionList<E> *l*, retorne una nueva lista que tenga los elementos de *l* en el mismo orden pero repetidos. Por ejemplo:

Si *l*=<a, b, c, d> el método deberá retornar *lNueva*=<a, a, b, b, c, c, d, d>

Para resolver este ejercicio deberá utilizar la sentencia for-each.

TIPIS:

- Antes de comenzar a desarrollar este trabajo práctico repase el diagrama de clases completo para visualizar todas las clases e interfaces que intervienen en la implementación del TDALista;
- Para programar el iterador utilice la estrategia de iterar sobre la estructura (no sobre una copia);
- Para que todo funcione correctamente utilice las interfaces iterable e iterator provistas por Java (`java.util`);
- Cada vez que resuelva un ejercicio considere si el ejercicio debe resolverse dentro de la estructura o fuera (como cliente).
- Es muy útil esbozar dibujos de las listas para ver como se deben modificar los enlaces ante determinadas situaciones.