Estructuras de Datos.
Facultad de Ciencias, UNAM 2015-2.
Práctica 05. Listas

Armando Ballinas Nangüelú.
armballinas@gmail.com

Fecha de entrega: 02 de abril de 2015

1. Introducción.

En esta práctica se implementarán varias versiones de listas. Las listas son una de las estructuras de datos más utilizadas y conocidas. Es por lo anterior que hay varias clases o tipos de listas y en esta práctica se trabajarán con algunos de ellos.

2. Conceptos adicionales.

Antes de poder implementar las listas serán necesarios algunos conceptos adicionales.

2.1. Genéricos en Java.

Dado que nos interesa tener implementaciones generales de listas homogéneas, es decir, que almacenen cualquier tipo de objeto y que todos los objetos que almacenen tengan el mismo tipo.

Una manera de lograr lo anterior en Java es utilizando el mecanismo del lenguaje conocido como genéricos.

En la práctica se necesitará hacer uso de genéricos tanto para la implementación como para la prueba de sus programas.

2.2. Paquetes en Java.

En la práctica se deben hacer uso de paquetes de forma obligatoria. Los paquetes en Java permiten encapsular de manera lógica aquellas clases e interfaces que tienen una característica en común. En nuestro caso utilizaremos un paquete para almacenar todas las implementaciones e interfaces de las estructuras de datos que creemos en el curso.

El paquete a utilizar se debe llamar `mx.unam.fciencias.EstDat`. Para hacer uso de este paquete, todos los archivos que implementen listas y las interfaces y excepciones relacionadas a ellas deben estar en el directorio ../mx/unam/fciencias/EstDat/ de su entrega, por ejemplo: src/mx/unam/fciencias/EstDat/.

Además todos estos archivos deberán tener como primera línea que no sea un comentario la instrucción: `package mx.unam.fciencias.EstDat;`. Esto para decirle a Java que el archivo pertenece al paquete en cuestión.

3. Tres listas, tres iteradores.

En la práctica se deben implementar tres listas diferentes, cada una con su iterador correspondiente:

1. Listas basadas en arreglos. Para implementar esta lista deben implementar en un archivo llamado `ListaArreglo.java`. En esta lista, los elementos se almacenan en un arreglo nativo de Java (`T[]` usando genéricos).Cuando los elementos ya no quepan en el arreglo usado, este se debe duplicar de tamaño automáticamente de manera que permita seguir almacenando elementos en la lista.

2. Listas ligadas simples. Esta lista se debe implementar en el archivo `ListaLigada.java`. En esta lista se usaran celdas para almacenar cada elemento de la lista. Recordemos que cada celda tiene el objeto que almacena y un apuntador a la siguiente celda. El apuntador de la última celda es nulo. Para implementar esta lista se debe implementar una clase privada llamada `Celda<E>` que representa una celda.

3. Listas doblemente ligadas. Esta lista se debe implementar en el archivo `ListaDoblementeLigada.java`. En esta lista, los elementos también se guardan en celdas, pero en este caso, cada celda tiene un apuntador a la celda anterior y otro a la celda siguiente. El apuntador a la celda anterior de la primera celda es nulo y el apuntador a la celda siguiente de la última también es nulo.

Cada implementación de Lista anterior debe implementar la interfaz `Lista<T>` en su totalidad, siguiendo las firmas de los métodos y documentación de la interfaz. En la interfaz se pide que cada lista sea capaz de devolver un `IteradorLista<T>`, por lo que para cada lista se debe crear una clase que implemente el iterador para dicha lista. En el caso de la lista doblemente ligada, su iterador debe implementar la interfaz `IteradorBidireccional<T>`.

4. Programa de prueba.

Además de las implementaciones de listas deben crear un programa de prueba en un archivo llamado `Main.java`. Este programa debe estar en el directorio raíz de su práctica y no debe pertenecer a ningún paquete. En este programa deben importar los archivos de las listas por ejemplo `import mx.unam.fciencias.EstDat.ListaArreglo`.

Su programa recibirá dos argumentos, el primero será una letra: A, L o D. Esta letrá indicará la representación de Listas a usar. En el caso de la A se usará una lista implementada con arreglos, en el caso de la L una lista ligada y en el caso de la D una lista doblemente ligada. Además como segundo argumento recibirá el nombre de un archivo de texto.

Este archivo de texto tendrá una serie de números, un número por línea.

Un ejemplo de la invocación del programa es: `java Main L nums.txt`

Su programa debe realizar lo siguiente:

1. Construir una lista del tipo especificado en el argumento.

2. Leer los números del archivo y almacenarlos en la lista.

3. Imprimir la lista en su orden actual a la salida estándar.

4. Invertir el orden de la lista es imprimirla de nuevo en el nuevo orden.

5. Imprimir el valor de la suma de todos los elementos de la lista.

6. Ordenar la lista en orden decreciente y volverla a imprimir.

5. Ejercicios.

1. (2 pts.) Realizar la implementación de ListaArreglo y de su iterador.

2. (3 pts,) Realizar la implementación de ListaLigada y de su iterador.

3. (3 pts.) Realizar la implementación de ListaDoblementeLigada y de su iterador.

4. (2 pts,) Realizar la implementación del programa de prueba Main.

6. Archivos adicionales.

Como parte de arhivos adicionales se incluyen las interfaces de los iteradores, la interfaz con el TDA Lista y una clase abstracta `ListaAbstracta.java` donde se implementan varios métodos comunes a todas las implementaciones de listas. También se añade una excepción `ExcepcionAccesoIncorrecto.j` que es una excepción lanzada por algunos métodos de las interfaces.

También se agregan dos archivos con números para que los usen como prueba.

7. Consideraciones de implementación y puntos extra

Se darán puntos extra a criterio del ayudante por buenas implementaciones de las listas e iteradores y por eficiencia en el programa de prueba.

1. Consideren implementar los iteradores como clases privadas de la lista en cuestión. No es necesario hacer lo anterior pero es una buena idea.

2. Traten de usar los iteradores lo más que puedan, sino, úsenlos al menos para imprimir las listas.

3. Dado que la fecha de entrega es en vacaciones, solo se tendrá una sesión de laboratorio para ver dudas. Tómenlo en cuenta.