

Programación II

PRÁCTICA **Secuencia Iterable**

DPTO. LSIIS. UNIDAD DE PROGRAMACIÓN

Normas y Objetivos

Objetivo: El objetivo de este ejercicio es la implementación de un TAD que proporcione el acceso secuencial a los elementos de una secuencia por medio de un cursor, y la posibilidad de modificar esta secuencia mediante operaciones de inserción y borrado basadas en la posición de dicho cursor.

Evaluación: La práctica se podrá realizar en grupos de dos alumnos o de forma individual. La nota que se obtenga en esta práctica tendrá un peso de 12,5 % en la nota final de la asignatura. Se mantendrá el mismo enunciado de la práctica para la convocatoria extraordinaria. La nota mínima de esta práctica para poder hacer media con el resto es de cuatro puntos. Como en el resto de prácticas de la asignatura, para que la nota se guarde hasta el examen extraordinario de Julio se debe sacar una nota mayor o igual que cinco.

Autoevaluación: El alumno debe comprobar que su ejercicio no contiene alguno de los errores explicados en el apartado 4 de este enunciado. **Si el ejercicio contiene alguno de estos errores, se calificará como suspenso.**

Grupos: Los grupos estarán formados por dos alumnos matriculados en el mismo semestre. Los alumnos deben estar matriculados y dados de alta en el sistema de entrega (maui) antes de proceder a realizar el registro del grupo. Ninguno de los dos alumnos deberá haber realizado entrega alguna de la práctica antes de definir el grupo. Una vez creado el grupo es el mismo para todas las prácticas en grupo del semestre. El alumno debe estar registrado en el sistema de entrega. El registro se hace una sola vez por semestre.

<http://maui.ls.fi.upm.es/entrega/formreg.php>

El grupo se crea por medio de la URL:

<http://maui.ls.fi.upm.es/entrega/CrearGruposProgramacion2.html>

El alumno de un grupo que realice la primera entrega de una práctica, será el que tenga que hacer todas las entregas para esa práctica.

Entrega: La práctica se entregará a través de la página web:

<http://maui.ls.fi.upm.es/entrega>

La práctica entregada debe compilar en la versión 1.6 del J2SE de Oracle/Sun y debe ser compatible con JUnit 4.8. En el momento de realizar la entrega, la práctica será sometida a una serie de pruebas que deberá superar para que la entrega sea admitida. El alumno dispondrá de un número máximo de diez entregas. Ahora bien, si el alumno realiza más de cinco entregas, se le restará un punto en la nota final de la práctica. Asimismo, por el hecho de que la práctica sea admitida, eso no implicará que la práctica esté aprobada. El fichero a entregar será `SecuenciaIterable.java`.

Código Auxiliar: La realización de esta práctica requiere la utilización de los ficheros auxiliares que se pueden encontrar dentro del fichero `cod_alumnosSecuencias.zip` que acompaña a este enunciado.

Fecha límite: el plazo límite es el día **31 de Mayo de 2012 a las 9:00 AM**.

Publicación de Notas: La publicación de las notas del ejercicio se anunciará a través del foro de prácticas.

Detección Automática de Copias: Cada práctica entregada se comparará con el resto de prácticas entregadas en todos los grupos de la asignatura. Esto se realizará utilizando un sofisticado programa de detección de copias.

Consecuencias de haber copiado: *Todos* los alumnos involucrados en una copia, bien por copiar o por ser copiados, quedan inhabilitados para presentarse a todas las convocatorias de examen del presente curso, además de la posible apertura de expediente académico.

1. Secuencia Iterable

Un TAD Secuencia Iterable es un TAD que permite crear, recorrer y modificar secuencias de elementos. Las operaciones que facilitan el recorrido y la modificación de la secuencia se realizan de distinta manera dependiendo de la posición de un cursor que forma parte del TAD.

Las operaciones para recorrer la secuencia tienen como objetivo proporcionar los servicios necesarios para mover el cursor a lo largo de la secuencia. De esta manera, el elemento al que se puede acceder directamente en un cierto momento es aquél sobre el que se encuentra el cursor. El cursor se puede desplazar una posición a la izquierda o a la derecha siempre que no se haya alcanzado todavía el principio o el final de la secuencia respectivamente.

En cuanto a las operaciones de modificación, se puede insertar o borrar un elemento con arreglo a la posición señalada por el cursor. En particular, hay dos operaciones de borrado. Una donde el elemento que se borra es el que señala el cursor y otra que elimina el elemento de la secuencia que se encuentra en una posición P relativa a la posición señalada por el cursor. Si la posición P es cero, entonces se borrará el elemento al que señala el cursor. Si la posición P es negativa, indica una posición que estará tantas veces a la izquierda del cursor como indique el valor de P . En cambio, si es positiva, indica una posición P que estará tantas veces a la derecha del cursor como indique el valor de P . Por otro lado, la inserción se puede llevar a cabo de dos maneras diferentes, insertando un elemento justo antes del elemento sobre el que se encuentra el cursor, o insertándolo justo después.

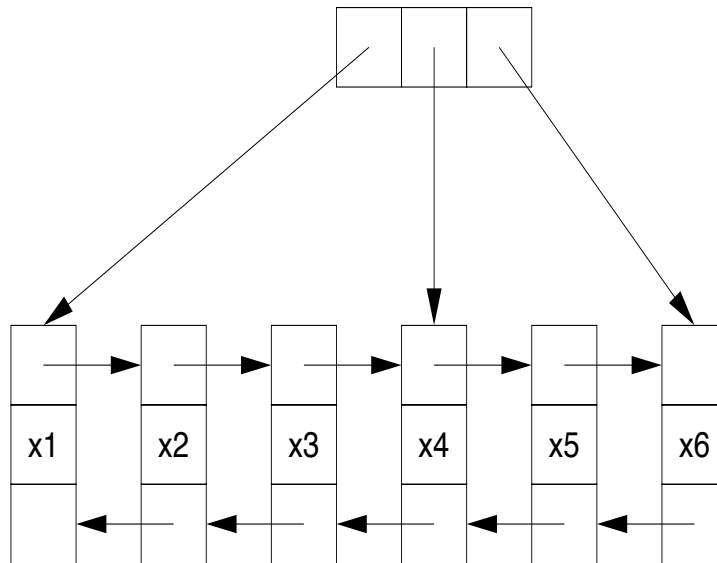
Por último, el TAD Secuencia Iterable también especifica su propio método `equals()` para determinar si dos secuencias son iguales. Dos secuencias son iguales si coinciden sus elementos desde el primero de la secuencia hasta el último. Esta operación no cambia la posición del cursor en ninguna de las dos secuencias.

2. Implementación de Secuencias Iterables

El alumno **deberá implementar en java el tipo abstracto de datos *Secuencia Iterable***. La especificación completa del tipo abstracto de datos `SecuenciaIterable` puede encontrarse en el fichero `SecuenciaIterableAbstracta.java`.

El alumno **deberá comprender y seguir la especificación junto con las restricciones de complejidad** para poder realizar la implementación (`SecuenciaIterable.java`).

La estructura de datos que se debe implementar es una cadena doblemente enlazada acompañada de tres punteros: uno al primer elemento o nodo de la secuencia, otro al último y otro representando el cursor. En la siguiente figura se puede ver un ejemplo de secuencia iterable con el *cursor* en el cuarto elemento:



3. Código proporcionado y fichero a entregar

Se deberá entregar el fichero `SecuenciaIterable.java`. Este fichero debe incluir la definición de la clase `SecuenciaIterable`. Esta clase se debe definir como una subclase (`extends`) de la clase `SecuenciaIterableAbstracta`, que se adjunta con este enunciado. Esto implica que la clase `SecuenciaIterable` debe implementar todos y cada uno de los métodos abstractos de la clase `SecuenciaIterableAbstracta`, así como el constructor. Asimismo, la clase `SecuenciaIterable` heredará los atributos declarados en `SecuenciaIterableAbstracta`, que deberán ser utilizados para implementar la cadena doblemente enlazada, más concretamente, los tres punteros que se mencionaron en el apartado anterior.

Junto a la clase `SecuenciaIterableAbstracta`, se proporciona el código de una clase auxiliar llamada `NodoDoble`, cuyo cometido es el de definir la estructura nodo de la cadena doblemente enlazada, y los métodos necesarios para trabajar con ella. Además, se proporcionan las clases que definen las excepciones que pueden lanzar los métodos de la clase `SecuenciaIterableAbstracta`.

Con el fin de probar el ejercicio, se proporciona un `jUnit` para la clase `SecuenciaIterable`.

4. Errores graves a evitar

En esta sección se enumeran algunos de los errores que el alumno debe evitar. Esta lista no es exhaustiva en el sentido de que no recoge todos los posibles errores que el alumno debe evitar cometer.

1. Se declaran atributos públicos.
2. Se realizan operaciones de entrada/salida excepto en las ramas del `catch` donde sí se permiten.
3. Se definen los atributos `primero`, `ultimo` y `actual` en la clase `SecuenciaIterable`.