

Laboratorio 3: Listas doblemente enlazadas/circulares

Importar a Eclipse el proyecto Lab3.

Ejercicio 1.

Sea la clase *MyOrderedLinkedList<T>*, que sirve para representar listas doblemente enlazadas, ordenadas, con apuntador al primer y último elemento.

```
public class MyOrderedLinkedList<T extends Comparable<T>> {  
    private Node<T> first;  
    private Node<T> last;  
    ....  
}
```

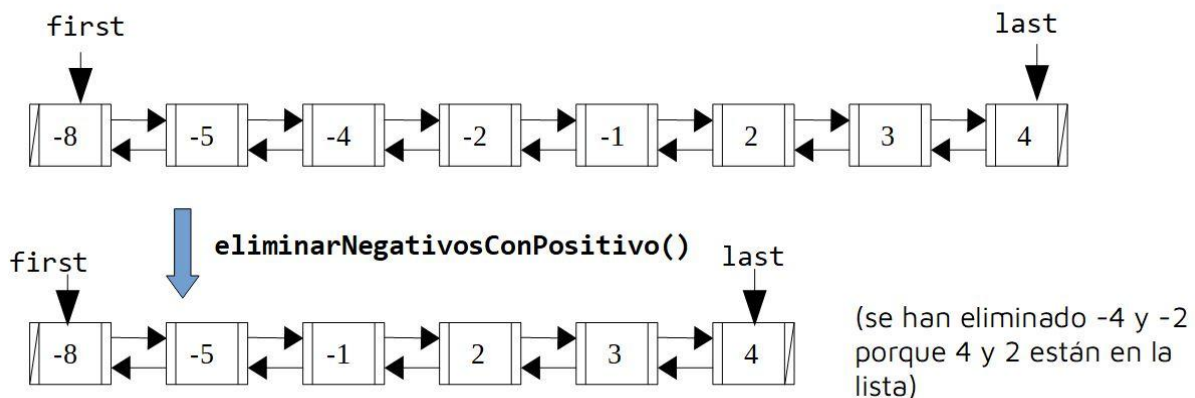
Se pide implementar el método *addOrdered* de dicha clase, que sirve para añadir un elemento de manera ordenada en dicha lista.

Ejercicio 2.

La clase *ListaEnteros* representa una lista de enteros, doblemente enlazada, con apuntador al primer y último elemento. La lista está ordenada y no contiene elementos repetidos.

Se pide implementar el método *eliminarNegativosConPositivo*. Este método elimina de la lista aquellos números negativos que tienen su correspondiente número positivo en la lista. El algoritmo debe ser de orden $O(N)$, siendo N el nº de elementos de la lista.

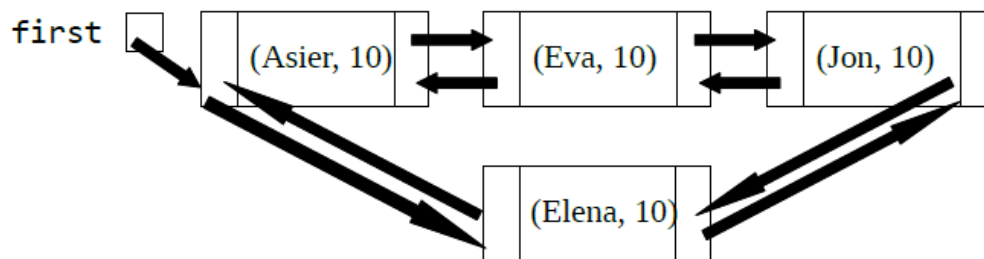
Ejemplo:



Ejercicio 3.

Queremos simular el comportamiento de una partida de dados con varios jugadores. Cada jugador parte con una puntuación inicial de 10 puntos. La partida se juega por turnos, y en cada turno, el jugador correspondiente tira el dado y se le resta de su puntuación el número obtenido en el dado. Cuando un jugador se queda sin puntos, queda eliminado de la partida. El ganador es el último jugador en quedarse con algún punto.

Para representar a la lista de jugadores de la partida vamos a utilizar una lista circular, doblemente encadenada y con apuntador al primer elemento.



Completar el método *jugar()*, que simula la partida:

- Simularemos el lanzamiento del dado mediante el método *lanzarDado*, que se os da implementado. Este método devuelve el nº de puntos (entre 1 y 6) que hay que restar al jugador actual.
- Cada vez que un jugador tire el dado se mostrará el nº que le ha salido y se actualizará su puntuación.
- Cuando un jugador quede eliminado porque se ha quedado sin puntos, se le borrará de la lista y se mostrará por pantalla el mensaje "X ha sido eliminado".
- Cuando sólo quede un jugador, se mostrará el mensaje "X es el ganador de la partida", y la partida finalizará.

Ejemplo de simulación (como la tirada del dado es aleatoria, os saldrán otros resultados):

ESTADO ACTUAL: Asier(10) Eva(10) Jon(10) Elena(10)

PULSA RETURN PARA TIRAR EL DADO:

Asier tira el dado: 2.

ESTADO ACTUAL: Asier(8) Eva(10) Jon(10) Elena(10)

PULSA RETURN PARA TIRAR EL DADO:

Eva tira el dado: 1.

ESTADO ACTUAL: Asier(8) Eva(9) Jon(10) Elena(10)

PULSA RETURN PARA TIRAR EL DADO:

Jon tira el dado: 2.

ESTADO ACTUAL: Asier(8) Eva(9) Jon(8) Elena(10)

PULSA RETURN PARA TIRAR EL DADO:

Elena tira el dado: 3.

ESTADO ACTUAL: Asier(8) Eva(9) Jon(8) Elena(7)

PULSA RETURN PARA TIRAR EL DADO:

Asier tira el dado: 4.

ESTADO ACTUAL: Asier(4) Eva(9) Jon(8) Elena(7)

PULSA RETURN PARA TIRAR EL DADO:

Eva tira el dado: 4.

ESTADO ACTUAL: Asier(4) Eva(5) Jon(8) Elena(7)

PULSA RETURN PARA TIRAR EL DADO:

Jon tira el dado: 6.

ESTADO ACTUAL: Asier(4) Eva(5) Jon(2) Elena(7)

PULSA RETURN PARA TIRAR EL DADO:

Elena tira el dado: 3.

ESTADO ACTUAL: Asier(4) Eva(5) Jon(2) Elena(4)

PULSA RETURN PARA TIRAR EL DADO:

Asier tira el dado: 6.

Asier ha sido eliminado.

ESTADO ACTUAL: Eva(5) Jon(2) Elena(4)

PULSA RETURN PARA TIRAR EL DADO:

Eva tira el dado: 4.

ESTADO ACTUAL: Eva(1) Jon(2) Elena(4)

PULSA RETURN PARA TIRAR EL DADO:

Jon tira el dado: 5.

Jon ha sido eliminado.

ESTADO ACTUAL: Eva(1) Elena(4)

PULSA RETURN PARA TIRAR EL DADO:

Elena tira el dado: 5.

Elena ha sido eliminado.

ESTADO ACTUAL: Eva(1)

Eva ha ganado la partida.