Examen de Febrero 2015

Nombre:_____**Especialidad**_____**Curso**____.

Una cola de doble entrada (también conocida como *deque*) es un tipo abstracto de datos que generaliza una cola, y en la que los elementos pueden ser añadidos o eliminados de la parte delantera o la parte posterior de la estructura. Las diferentes operaciones que ofrece esta estructura de datos son:

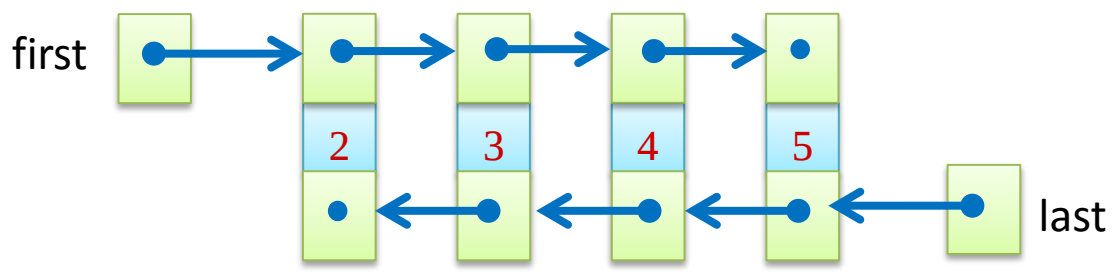
- **`empty`** : devuelve una nueva cola de doble entrada sin elementos.
- **`isEmpty`** : comprueba si una cola de doble entrada está vacía.
- **`addFirst`** : añade un nuevo elemento en la parte delantera de la cola.
- **`addLast`** : añade un nuevo elemento en la parte posterior de la cola.
- **`first`** : devuelve (sin eliminar) el elemento situado en la parte delantera de la cola.
- **`last`** : devuelve (sin eliminar) el elemento situado al final de la cola.
- **`deleteFirst`** : elimina el elemento situado en la parte delantera de la cola.
- **`deleteLast`** : elimina el elemento situado al final de la cola.

Implementación en Java

Una cola de doble entrada puede implementarse en Java mediante una estructura de datos lineal doblemente enlazada. Los nodos de esta estructura consisten en tres componentes:

- El elemento almacenado en el nodo,
- Una referencia al siguiente nodo en la estructura doblemente enlazada (**`null`** si el nodo actual es el último en la estructura),
- Una referencia al nodo anterior en la estructura doblemente enlazada (**`null`** si el nodo actual es el primero en la estructura).

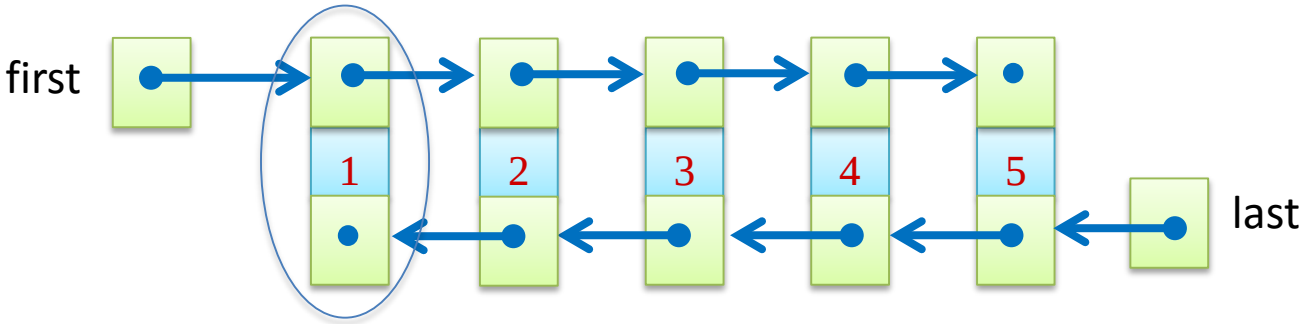
Se utilizarán dos variables de instancia (**`first`** y **`last`**) como referencias al primer y último nodo de la estructura de datos. Por ejemplo, la siguiente figura muestra la representación de una cola de doble entrada de enteros con 4 elementos:



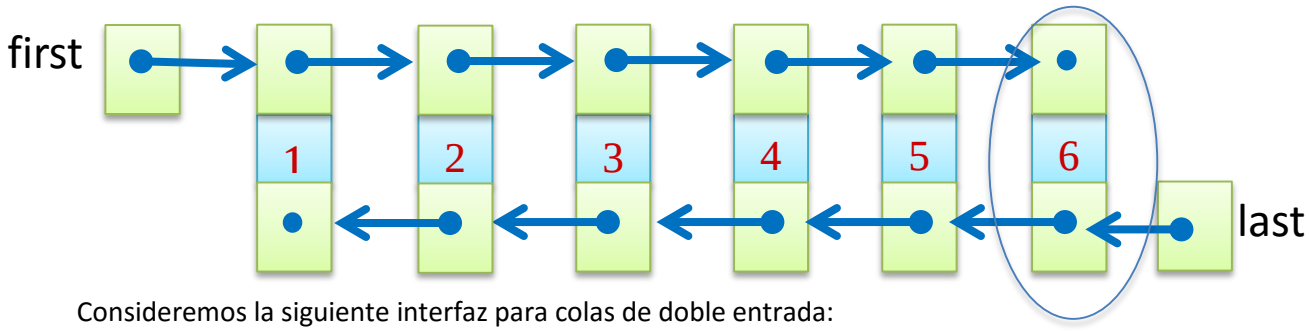
donde 2 es el primer elemento (el elemento en la parte delantera) y 5 es el último (el elemento en la parte posterior).

Examen de Febrero 2015

La operación **addFirst** debe crear un nuevo nodo con el elemento a añadir y enlazarlo al principio de la estructura de datos. La cola resultante después de la adición del elemento 1 en la parte delantera de la cola anterior se muestra en la siguiente figura:



La operación **addLast** debe crear un nuevo nodo con el elemento a añadir y enlazarlo al final de la estructura de datos. La cola resultante después de la adición del elemento 6 en la parte posterior de la cola anterior se muestra en la siguiente figura:



Consideremos la siguiente interfaz para colas de doble entrada:

```
package dataStructures.doubleEndedQueue;

public interface DoubleEndedQueue<T> {
	boolean isEmpty();
	void addFirst(T x);
	void addLast(T x);
	T first();
	T last();
	void deleteFirst();
	void deleteLast();
}
```

y la plantilla **LinkedDoubleEndedQueue** disponible en el campus virtual para implementar las colas de doble entrada, como se ha descrito.

Vuestro trabajo consiste en:

1. completar la plantilla con los diferentes métodos de la clase y
2. determinar la complejidad asintótica de las diferentes operaciones (incluir vuestro análisis como comentarios en cada operación dentro de la propia plantilla junto a la solución).

Examen de Febrero 2015

Implementación en Haskell

Una cola de doble entrada se puede representar en Haskell mediante una estructura que encapsule dos listas:

```
data DEQue a = DEQ [a] [a]
```

Los elementos de la primera lista (del primero al último) corresponden con los primeros elementos de la cola y los elementos de la segunda lista corresponden con los elementos de la parte posterior de la cola, pero estos están almacenados en orden inverso. Por ejemplo, una cola de doble entrada que almacena los elementos (2, 3, 4, 5, 6 y 7) de manera que 2 es el elemento situado en la parte delantera de la cola y 7 en la parte posterior se puede representar como:

```
DEQ [2,3,4] [7,6,5]
```

Nótese que con esta representación, la cabeza de la primera lista es el primer elemento de la cola mientras que la cabeza de la segunda es el último elemento.

Si se inserta un nuevo elemento 1 en la parte delantera de la cola, la representación resultante sería :

```
DEQ [1,2,3,4] [7,6,5]
```

Y si se inserta el elemento 8 en la parte posterior de la cola, la representación de la cola resultante sería:

```
DEQ [1,2,3,4] [8,7,6,5]
```

Para acceder (o eliminar) un elemento de la parte delantera de la cola, debemos en primer lugar comprobar si la primera lista está vacía; si no lo está, se debe devolver el elemento situado en la cabeza de la primera lista (o eliminar). Por ejemplo, si se elimina un elemento de la parte frontal de la cola anterior, la representación se convierte en:

```
DEQ [2,3,4] [8,7,6,5]
```

Si tenemos que eliminar un elemento de la parte delantera de la cola y la primera lista está vacía pero la segunda no lo está, se deberá dividir la segunda lista en dos mitades iguales (si la longitud es impar, la segunda mitad será mayor). La primera mitad se convierte en la nueva segunda lista y la segunda mitad se debe invertir y se convierte en la nueva primera. Después de hacer este ajuste, el primer elemento estará de nuevo situado a la cabeza de la primera lista. Por ejemplo, si quitamos 3 elementos de la parte frontal de la cola:

```
DEQ [2,3,4] [8,7,6,5]
```

La representación pasaría a ser:

```
DEQ [] [8,7,6,5]
```

Examen de Febrero 2015

Si ahora tratamos de eliminar un elemento más de la parte delantera, la primera lista está vacía. En este caso, la lista **[8,7,6,5]** se divide en **[8,7]** y **[6,5]**, **[6,5]** se invierte y la representación se convierte en:

```
DEQ [5,6] [8,7]
```

Después de este ajuste, la primera lista es no vacía y contiene al menos la mitad de los elementos en la cola, por lo que el primer elemento de la cola está de nuevo a la cabeza de la primera lista y puede ser devuelto o eliminado fácilmente.

El procedimiento para acceder o eliminar un elemento de la parte posterior de la cola en caso de que la segunda lista esté vacía es simétrico al que acabamos de describir, pero ahora sería la primera lista la que habría que dividir.

Vuestro trabajo consiste en:

1. completar la implementación de una cola de doble entrada como se ha descrito previamente completando la plantilla **DoubleEndedQueue** disponible en el campus virtual.
2. determinar la complejidad asintótica de las diferentes operaciones (incluir vuestro análisis como comentarios en cada operación dentro de la propia plantilla junto a la solución).