

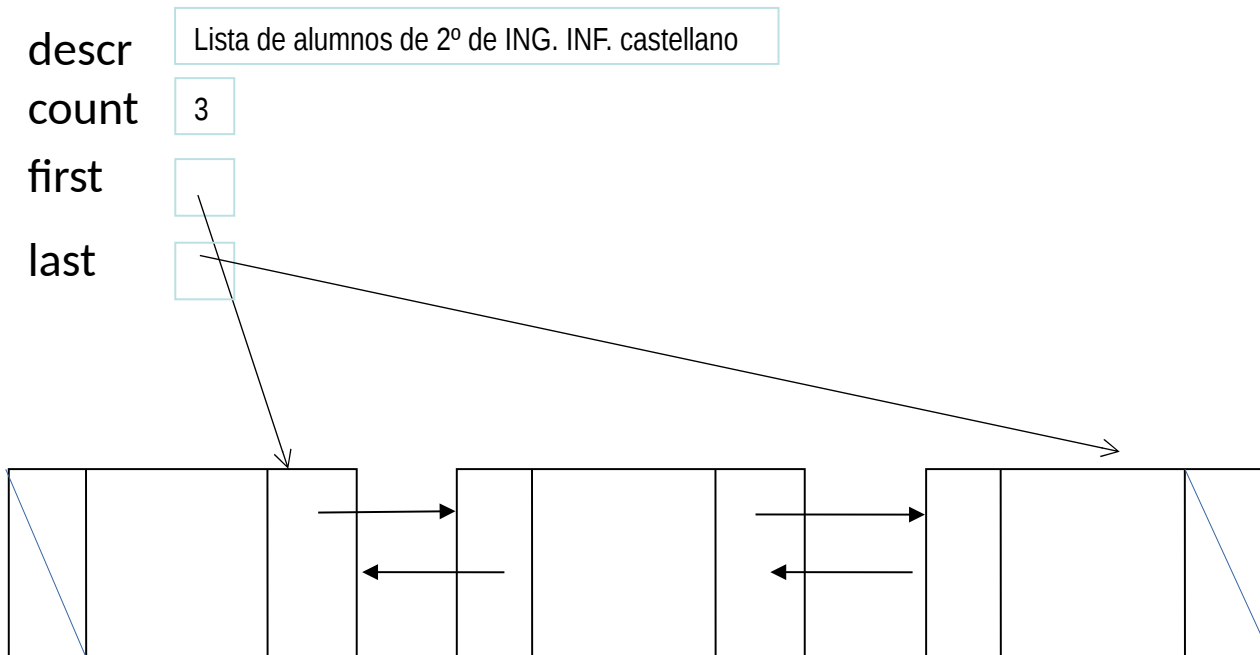
Estructuras de Datos y Algoritmos

Laboratorio estructuras enlazadas

Diseño de una estructura enlazada

- Especificar, diseñar e implementar en Java los métodos de la clase `DoubleLinkedList` que figuran en la tabla adjunta. Se pide también el programa de pruebas que se haya diseñado y la complejidad de cada método.

Estructura



Métodos y Descripción. TAD Lista (ListADT): operaciones comunes a todas las listas

- **public void setDescr(String nom);** // Añade el nombre como descripción de la lista
- **public String getDescr();** // Devuelve la descripción de la lista
- **public T removeFirst();** // Elimina el primer elemento de la lista
- **public T removeLast();** // Elimina el último elemento de la lista
- **public T remove(T elem);** // Elimina un elemento concreto de la lista (primera aparición)
- **public void removeAll(T elem);** // Elimina todas las apariciones de un elemento de la lista
- **public T first();** // Da acceso al primer elemento de la lista
- **public T last();** // Da acceso al último elemento de la lista
- **public ListADT<T> clone();** // Devuelve una copia de la lista
- **public boolean contains(T elem);** // Determina si la lista contiene un elemento concreto
- **public T find(T elem);** // Determina si la lista contiene un elemento concreto, y devuelve su referencia, null en caso de que no esté
- **public boolean isEmpty();** // Determina si la lista está vacía
- **public int size();** // Determina el número de elementos de la lista
- **public Iterator<T> iterator();** // Devuelve un iterador a los elementos de la lista

Métodos y Descripción.

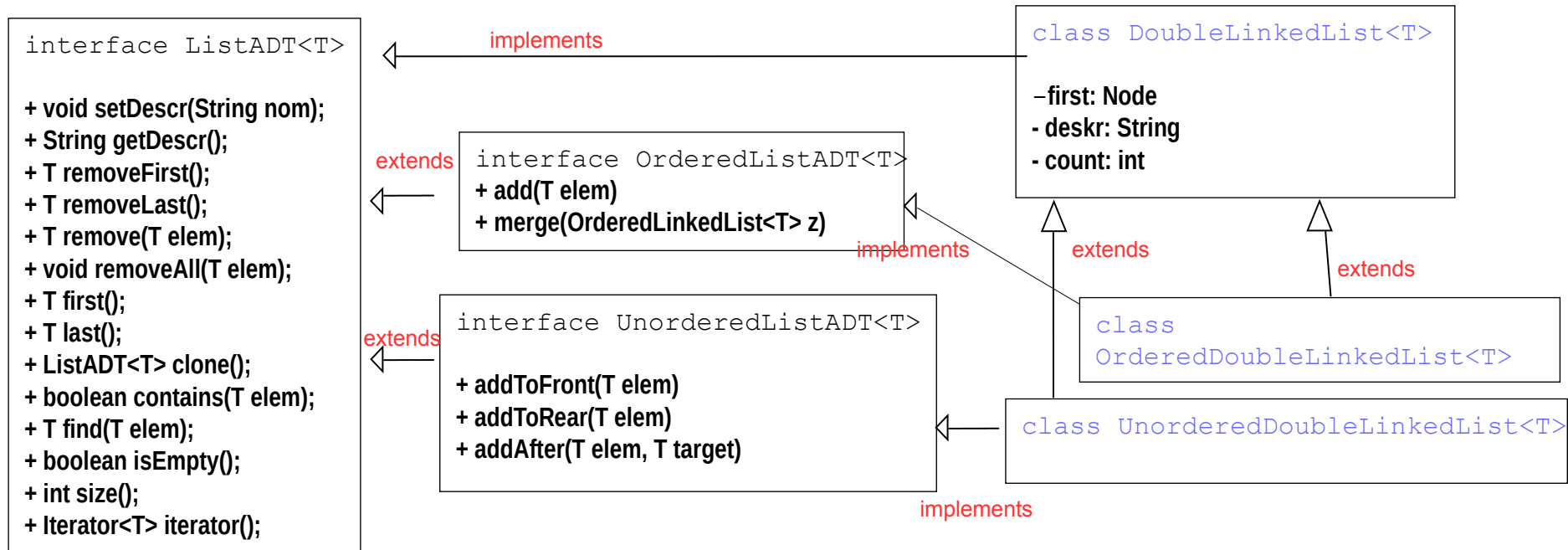
TAD Lista desordenada (UnorderedListADT): operaciones de listas desordenadas:

- **public void addToFront(T elem);**
▪ // añade un elemento al comienzo
- **public void addToRear(T elem);**
▪ // añade un elemento al final
- **public void addAfter(T elem, T target);**
▪ // Añade elem detrás de otro elemento concreto, target, que ya se encuentra en la lista

TAD Lista ordenada (OrderedListADT): operaciones de listas ordenadas:

- **public void add(T elem);**
▪ // añade el elemento en su posición
- **public void merge(OrderedList<T> lista);**
// se ha fusionado la lista con los elementos de la lista actual (debe ser $O(n)$)

Diagrama de clases



Tareas a realizar

- Especificar, diseñar e implementar en Java los métodos de las clases `DoubleLinkedList` y `UnorderedDoubleLinkedList` que figuran en la tabla adjunta. Se pide también el programa de pruebas que se haya diseñado y la complejidad de cada método.
- Se deberá sustituir la nueva clase `UnorderedDoubleLinkedList` en alguna de las listas usadas en la fase 1 de la práctica, y comprobar que funciona correctamente (por ejemplo, en la lista de enlaces salientes desde una web)
- La implementación de la clase `OrderedDoubleLinkedList` es opcional