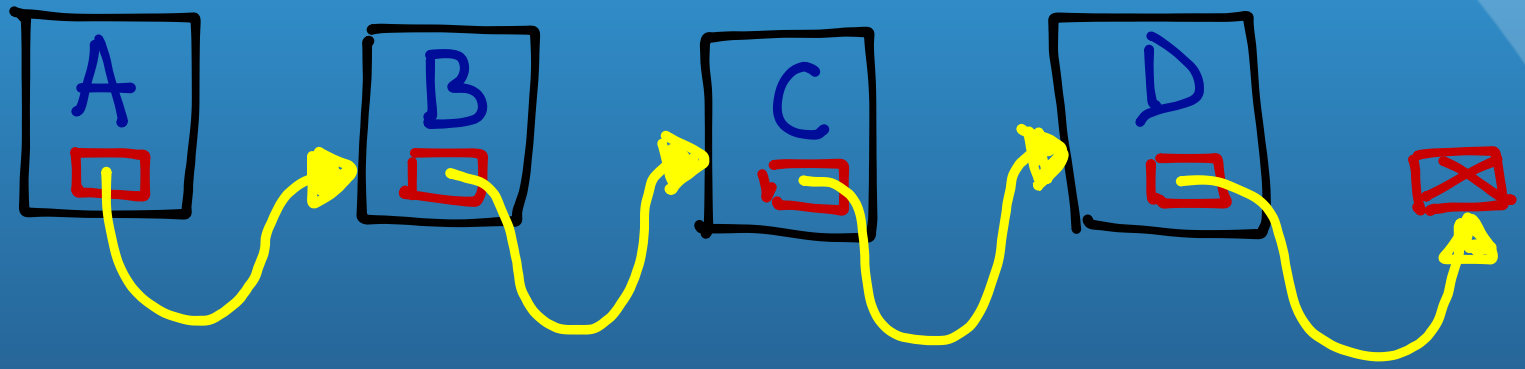
Estructuras de Datos IIS262

Profesor: Patricio Galeas

CAPÍTULO 5 : Listas Enlazadas



Introducción

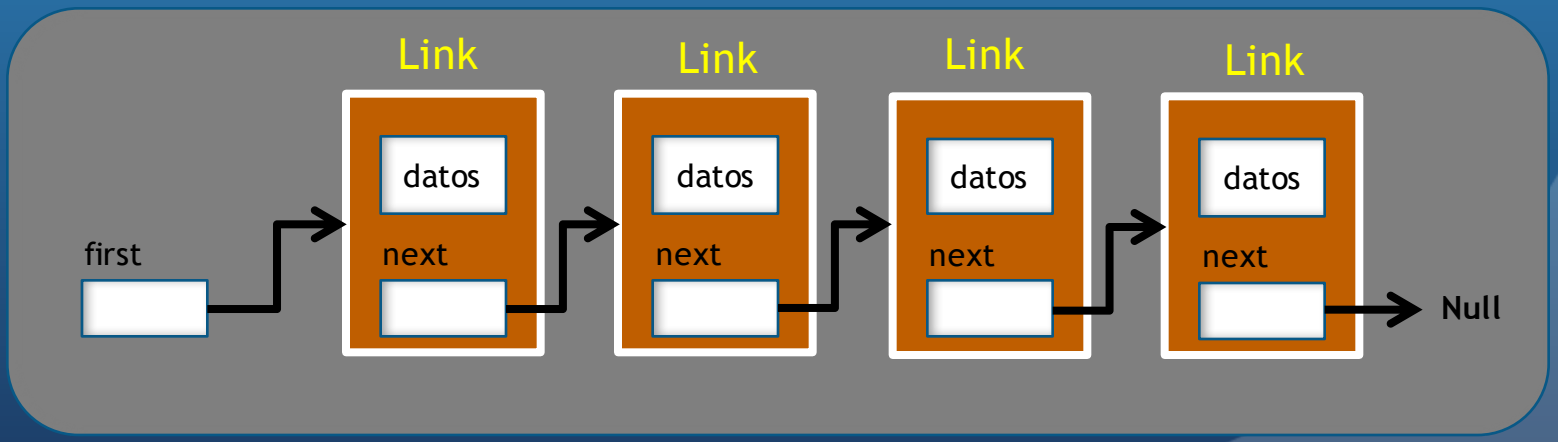
- Hemos visto
  - Arreglo desordenado : búsqueda
  - Arreglo ordenado : inserción
  - Ambos arreglos : eliminación

  } lenta
  - El tamaño no puede ser cambiado después de creado el arreglo.

- Las Listas Enlazadas (LE) resuelven muchos de estos problemas
- Son útiles en:
  - Bases de datos
  - Como reemplazo de arreglos en colas y pilas
- Son relativamente simples (menos complicados que los árboles)
- Sin embargo, también tienen algunas restricciones.

¿Que son las Listas Enlazadas?

- En una Lista Enlazada cada ítem esta inmerso en un link.
- Un link es un objeto de una clase llamada por ejemplo Link.
- Cada objeto link tiene una referencia (llamada usualmente next) al siguiente link de la lista.
- Un campo en la misma lista contiene una referencia a la primer link.



¿Que son las Listas Enlazadas?

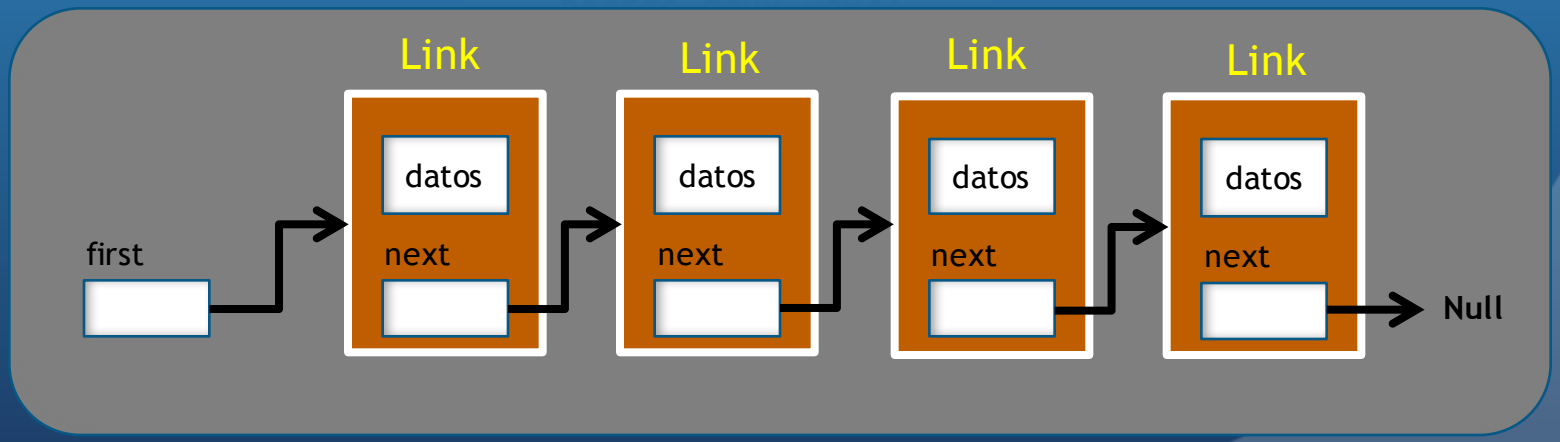
- Parte de la definición de la clase Link:

```
class Link
   {
   public int iData;
   public double dData;
   public Link next;
   }
```

Dato

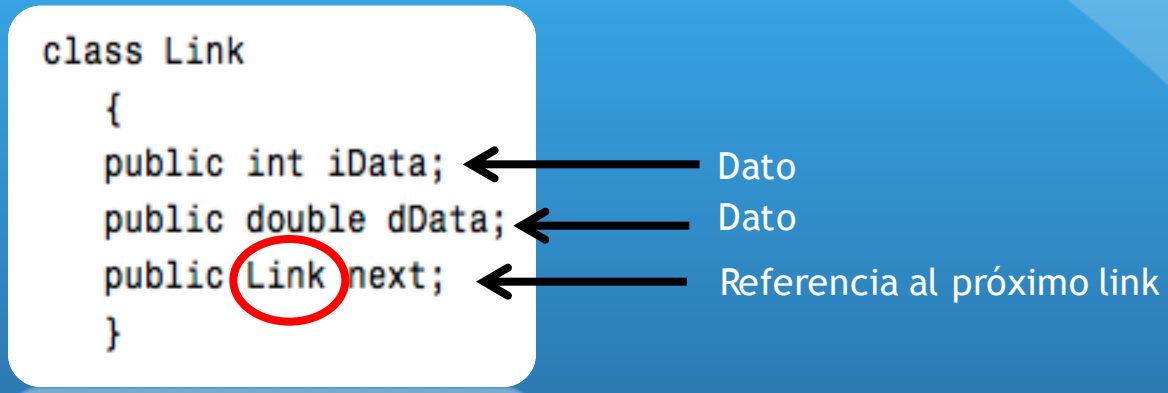
Dato

Referencia al próximo link



¿Que son las Listas Enlazadas?

- La clase Link se referencia a si misma: ¿puede ser un problema?

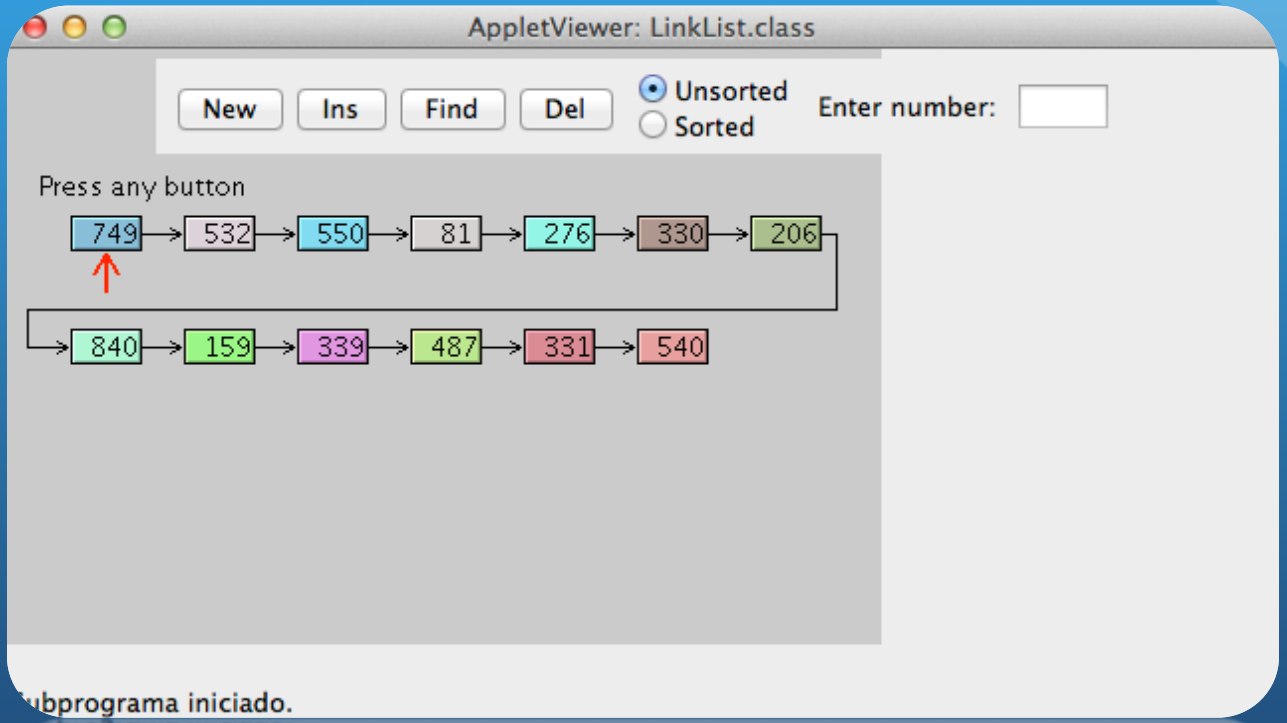


- En Java un objeto Link no contiene realmente otro objeto Link, sino que una referencia de este último.
- Una referencia es la dirección del objeto en la memoria del computador

Arreglos vs Listas Enlazadas

- En un arreglo cada ítem ocupa una posición particular. Esta posición puede ser accedida directamente usando el índice.
- En una lista, la única forma de encontrar un ítem es siguiendo la secuencia en la cadena de ítems.

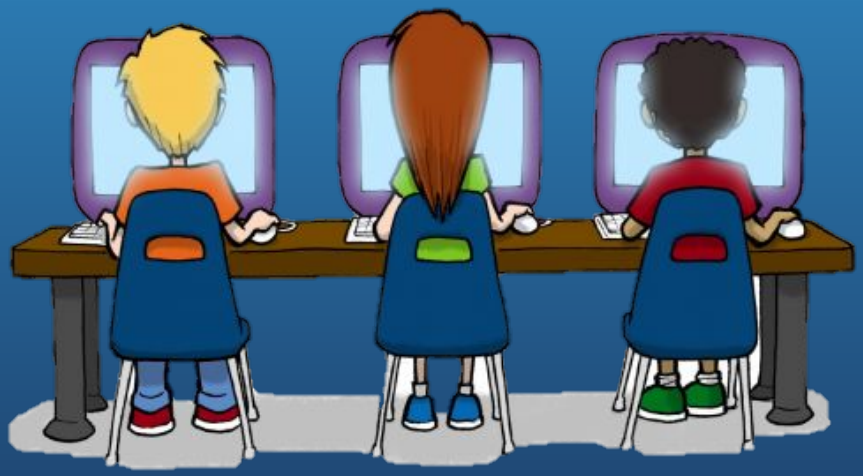
Funcionamiento de una Lista Enlazada



Implementación de una Lista Enlazada Simple en Java

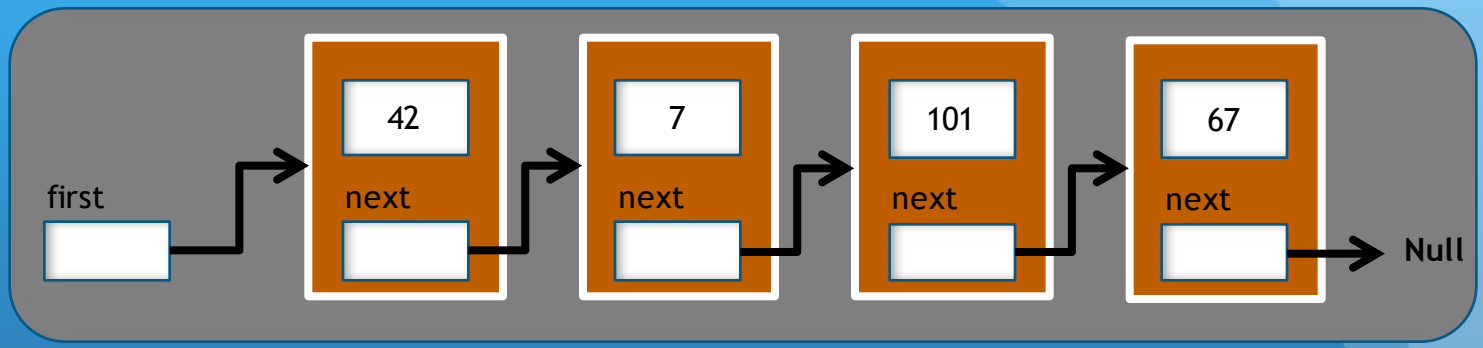
linklist.java

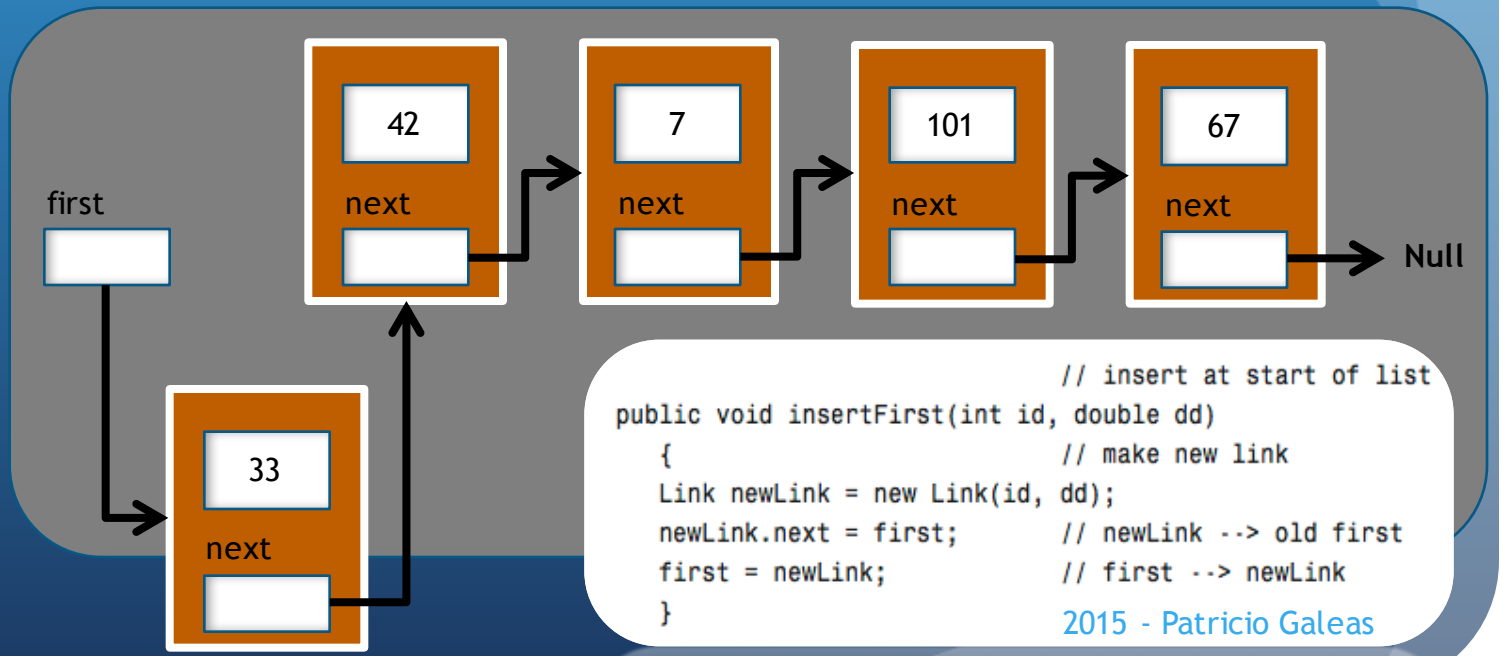
code



Implementación de una Lista Enlazada Simple en Java

Insertando un link



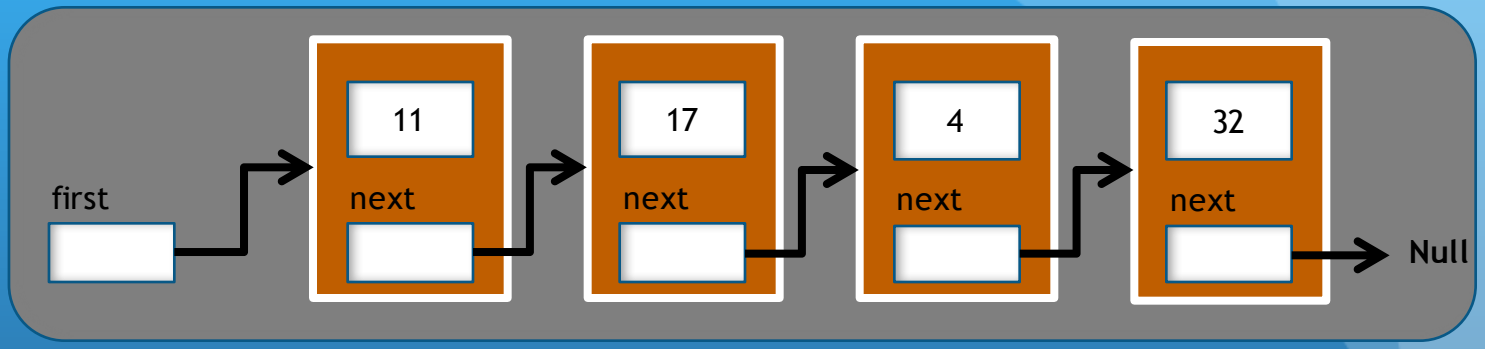



```
                                    // insert at start of list
public void insertFirst(int id, double dd)
   {                                // make new link
   Link newLink = new Link(id, dd);
   newLink.next = first;            // newLink --> old first
   first = newLink;                 // first --> newLink
   }
```

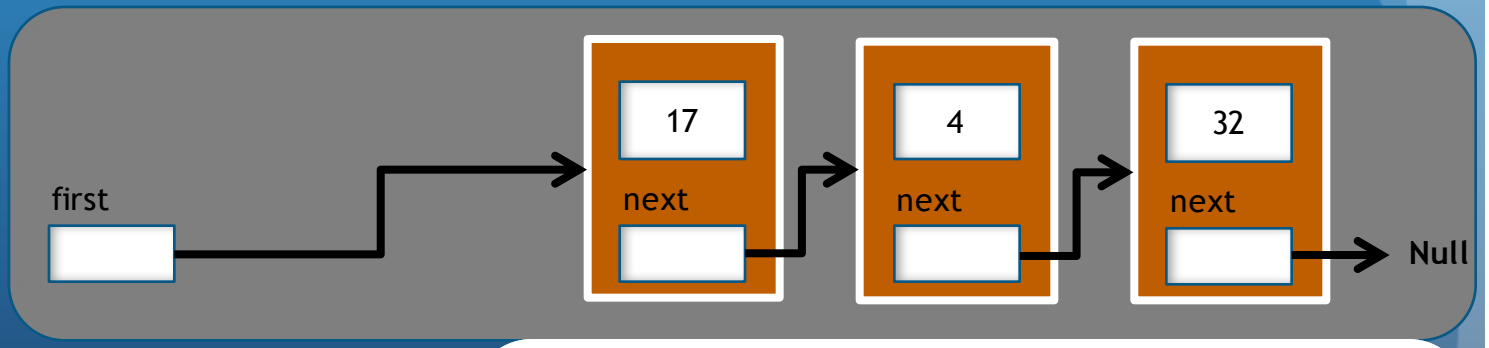
Implementación de una Lista Enlazada Simple en Java

Eliminando el primer link

antes



después



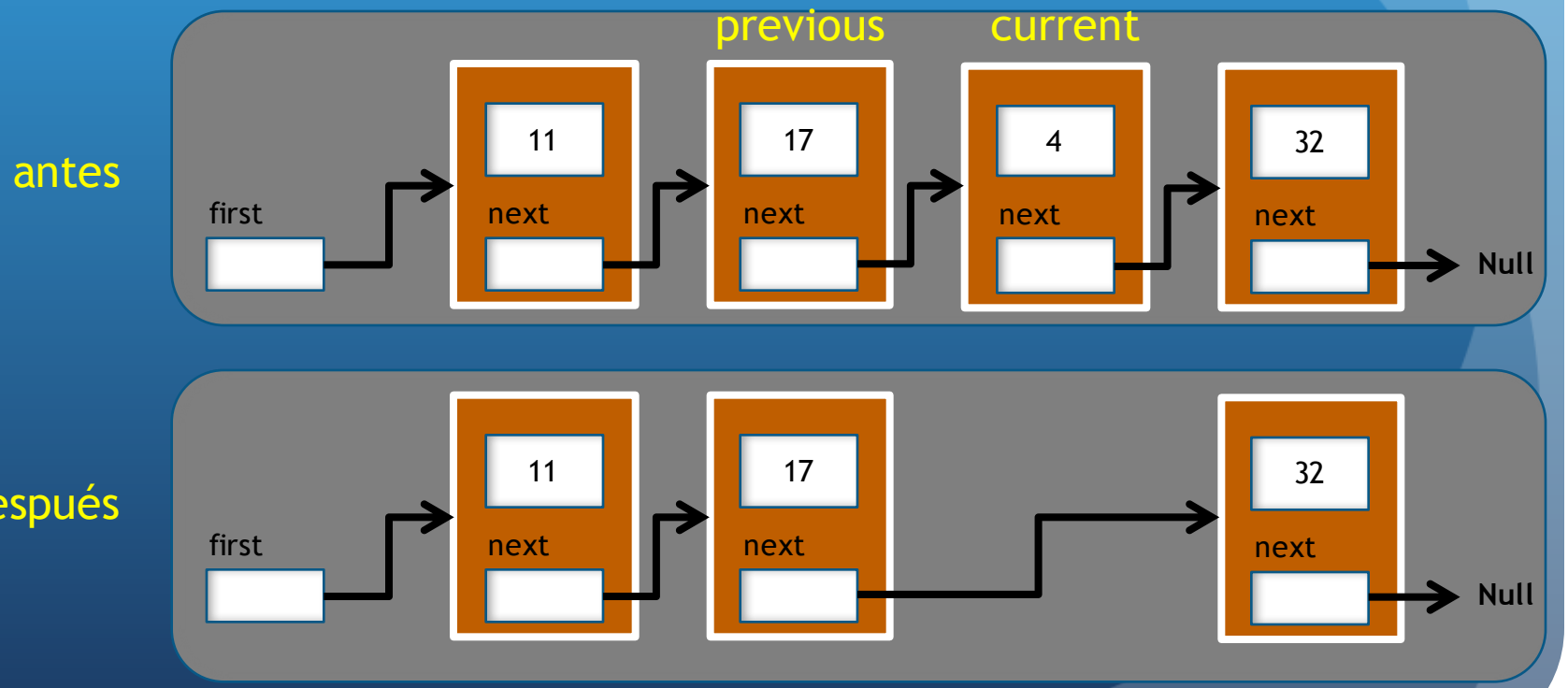

```
public Link deleteFirst()       // delete first item
   {                            // (assumes list not empty)
   Link temp = first;           // save reference to link
   first = first.next;          // delete it: first-->old next
   return temp;                 // return deleted link
   }
```

Buscando y Borrando Links

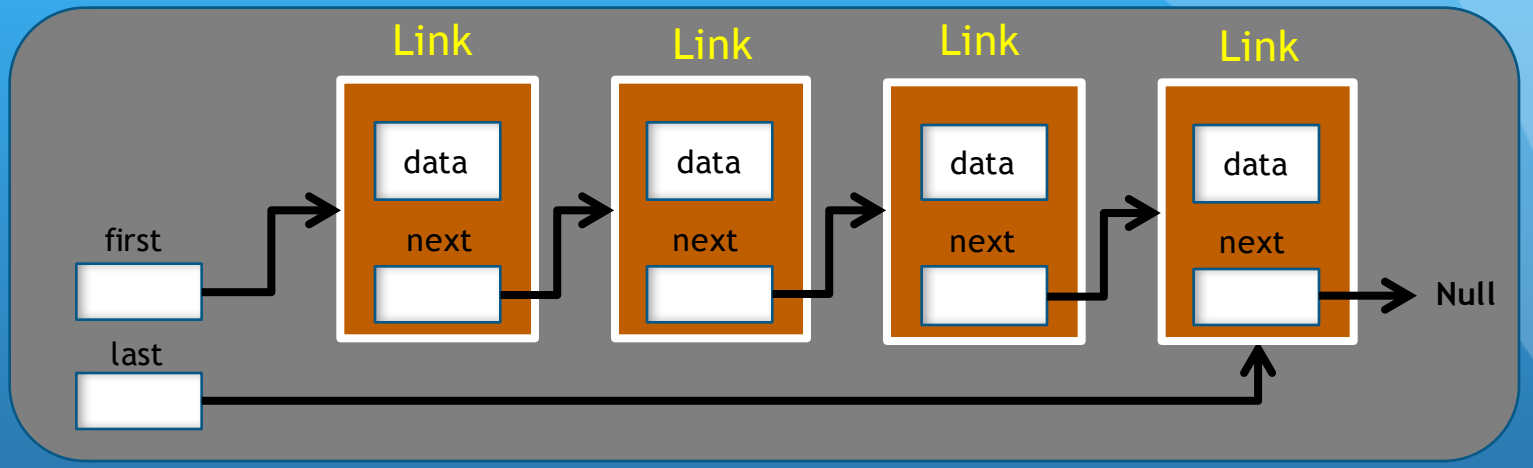
- Para buscar y borrar Links, se agregan dos métodos a la rutina anterior.
- En linkList2.java, se incluye find() y delete()

code

Eliminando un link cualquiera



Listas Doblemente Terminadas

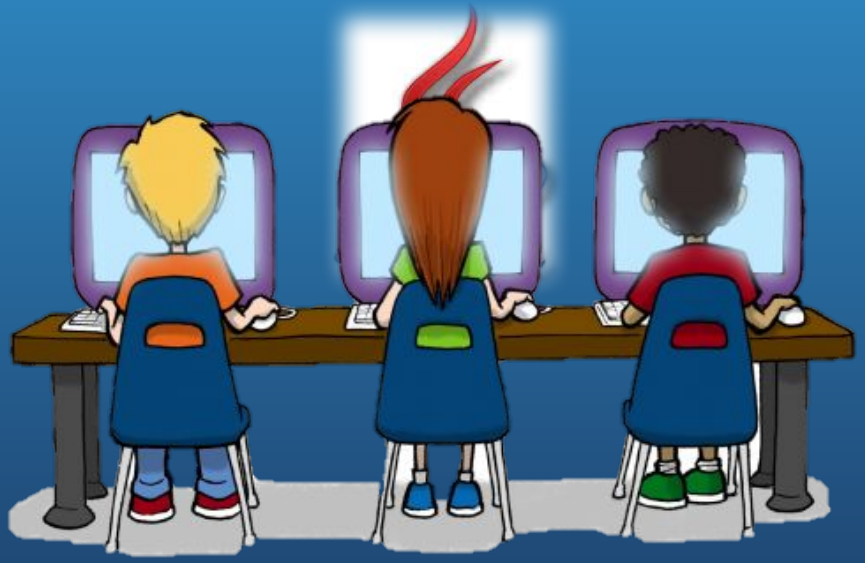


- Cuenta con un enlace más al último elemento. Se puede insertar directamente el último elemento.
- Esta estructura es muy útil para implementar una cola.

Implementación de una Lista Doblemente Terminada en Java

firstLastList.java

code

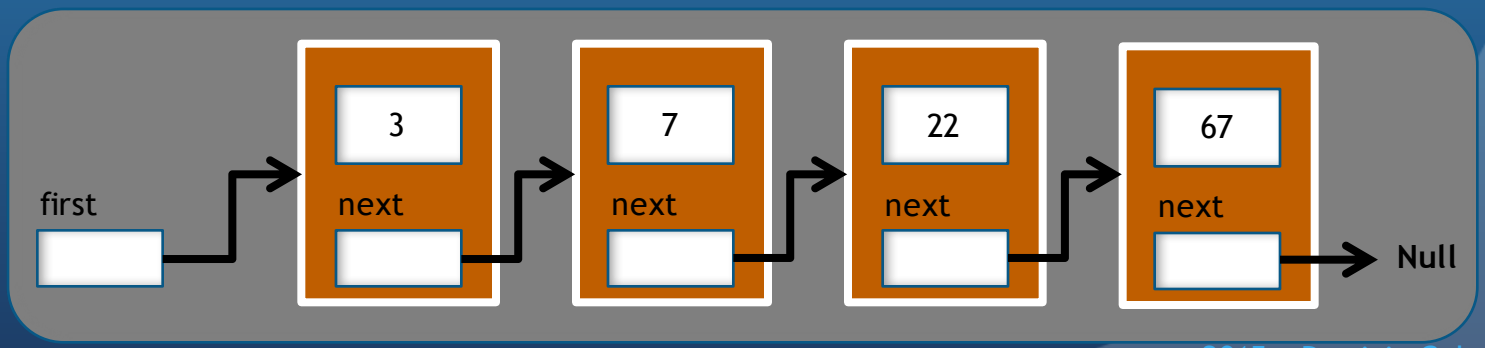


Eficiencia de las Listas Enlazadas

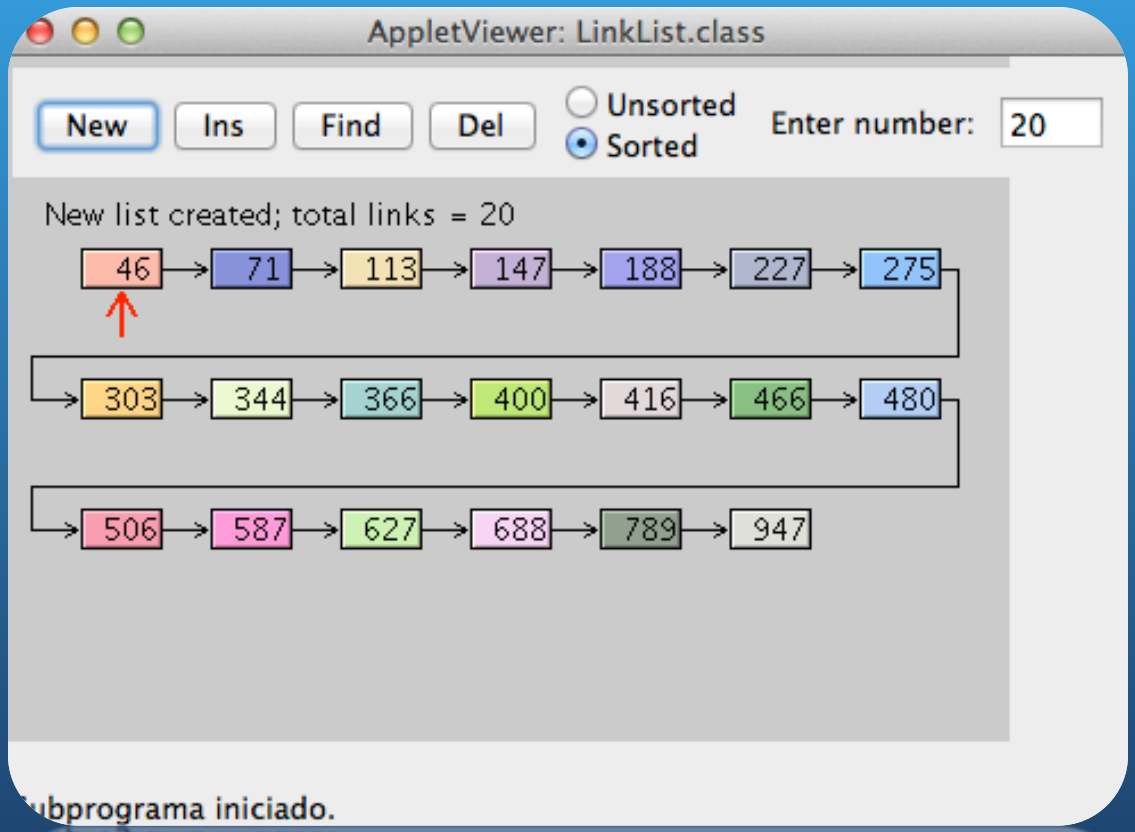
- La inserción y eliminación al comienzo de una lista enlazada es muy rápido. Implica cambiar uno o dos referencias : O(1)
- Encontrar, borrar o insertar un elemento en una cierta posición requiere buscar en promedio en la mitad de los ítems de la lista: O(N) comparaciones. Pero no necesita mover para insertar o borrar (más rápido que el arreglo).
- Otra ventaja sobre arreglos, es que la lista enlazada ocupa sólo la memoria que va necesitando (no es necesario definir su tamaño previamente).

Listas Ordenadas

- Los elementos están ordenados según su valor.
- Una lista ordenada se usa en situaciones similares en las que se aplica el arreglo ordenado.
- Lista Ordenada vs Arreglo Ordenado
  - La lista es más rápida en la inserción (no mueve elementos).
  - La lista puede crecer en forma dinámica.
  - La lista es más difícil de implementar.



Funcionamiento
de una Lista Ordenada



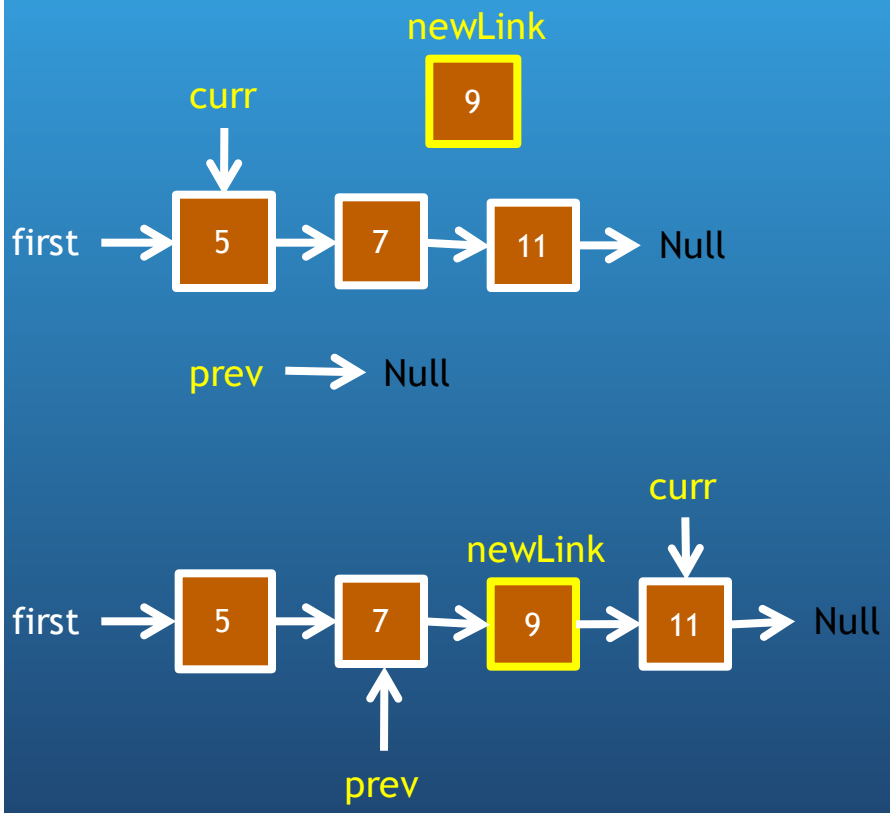
applet

sortedList.java

code

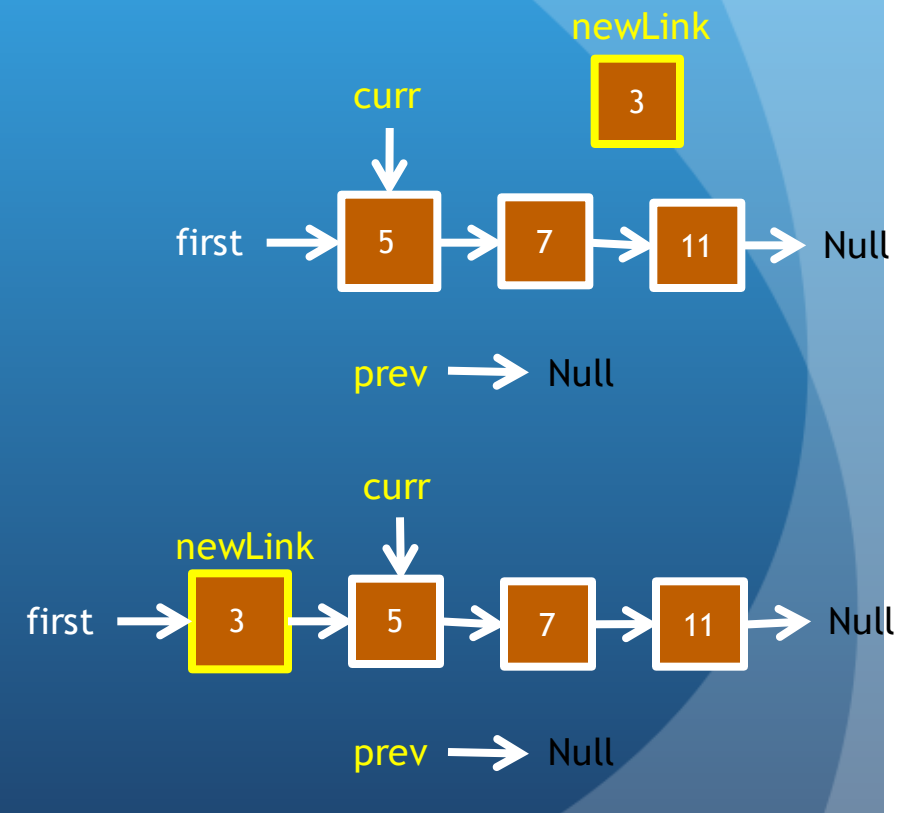
Insertando en una Lista Ordenada

CASO 1 : newLink no va al principio



newLink.next = curr
Prev.next = newLink

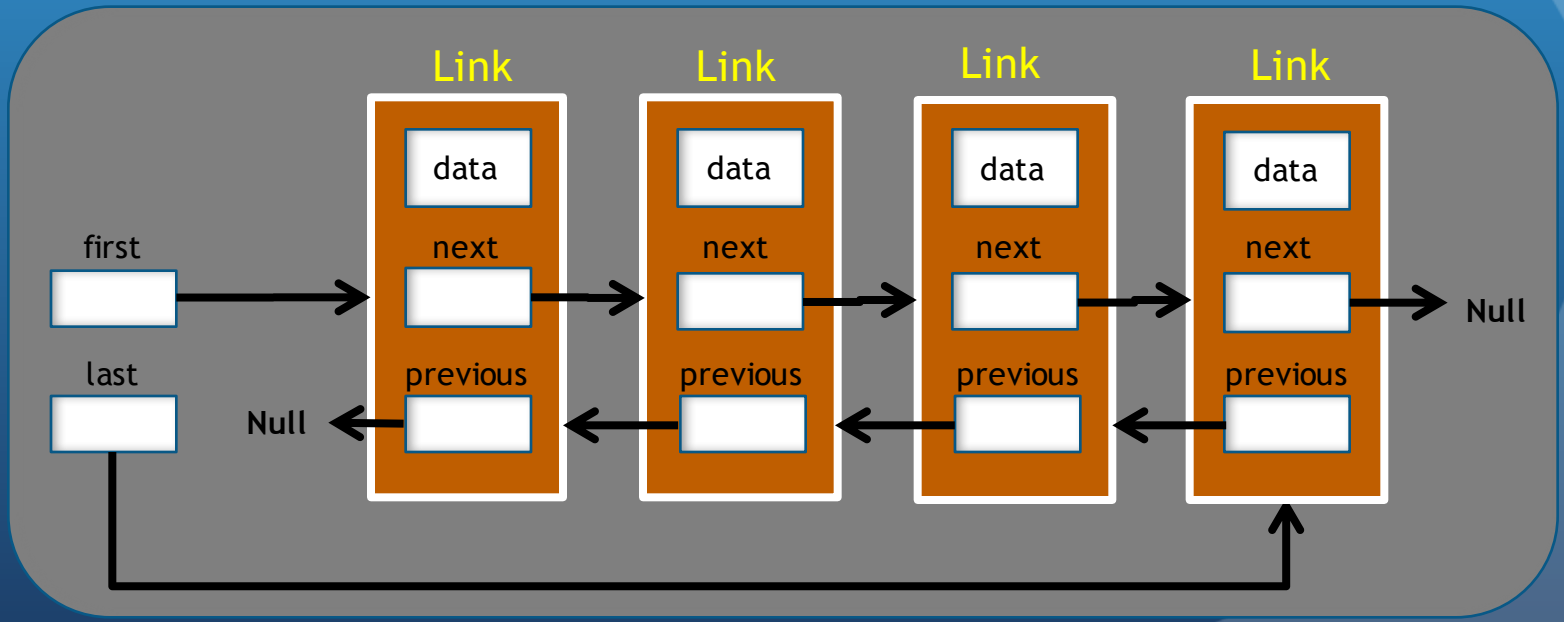
CASO 2 : newLink va al principio



newLink.next = curr
first = newLink

Listas Doblemente Enlazadas

- Cada Link poseen dos referencias, una al antecesor y otra al sucesor
- Permiten recorrer la lista en las dos direcciones.
- Para insertar o borrar un Link, es necesario manejar 4 links.

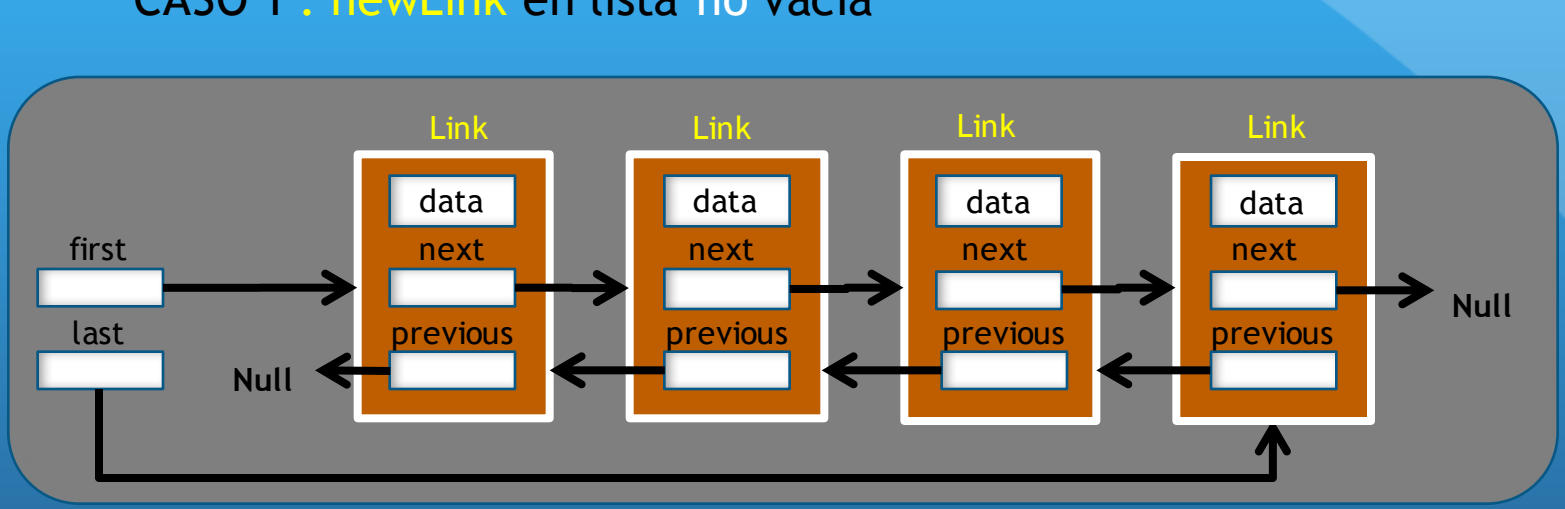


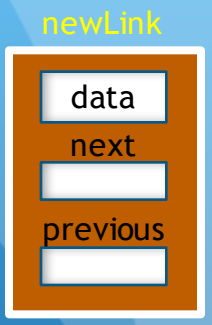
Insertando al principio
Lista Doblemente Enlazada

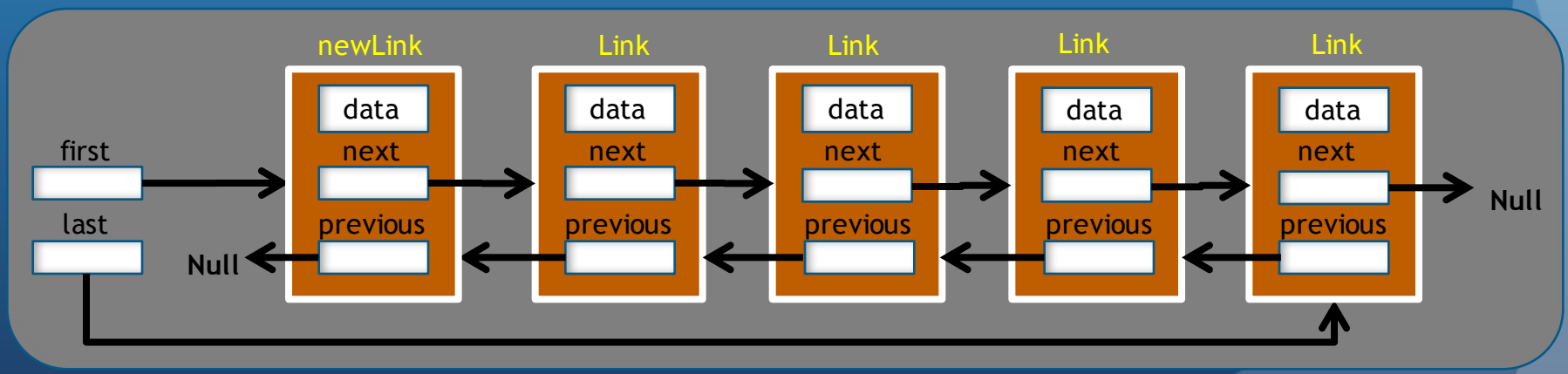
doublyLinked.java

code

CASO 1 : newLink en lista no vacía







newLink.next = first

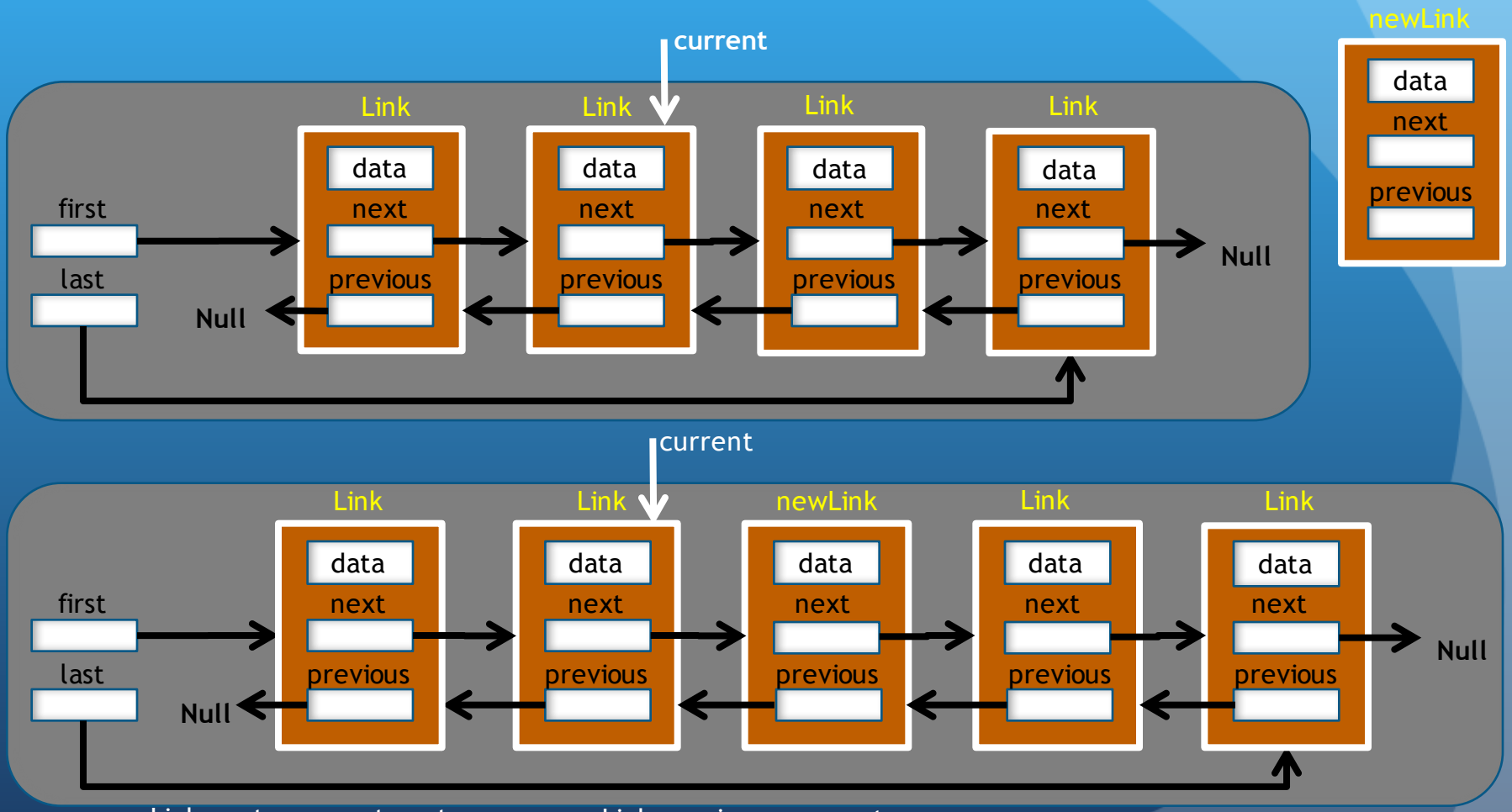
first.previous = newLink
first = newLink

Insertando después de ...
Lista Doblemente Enlazada

doublyLinked.java

code

CASO 1 : newLink no se inserta al final



newLink.next = current.next
current.next.previous = newLink

newLink.previous = current
current.next = newLink

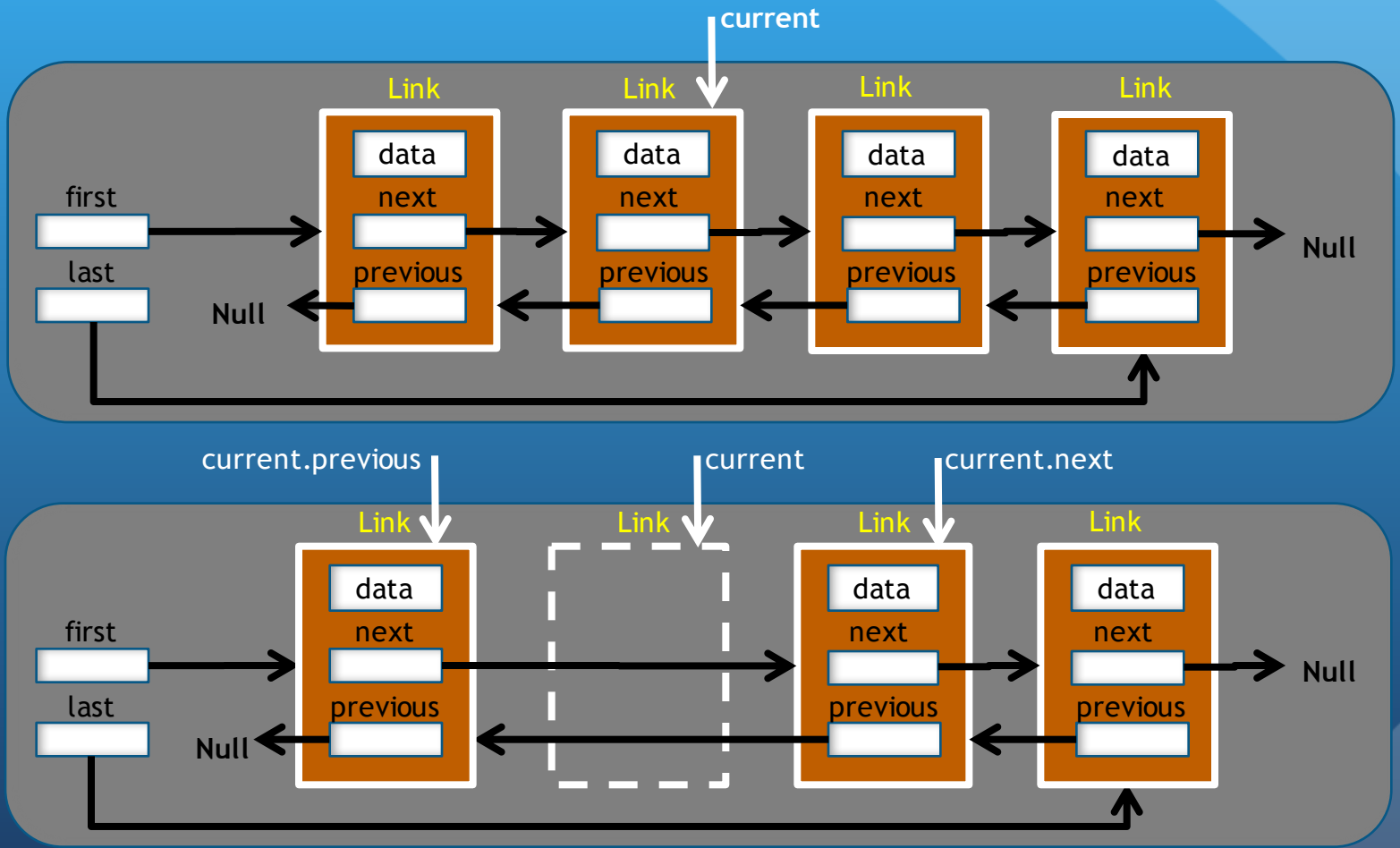
doublyLinked.java

code

Eliminando ítem

Lista Doblemente Enlazada

CASO 1 : eliminando ítem con valor específico en medio de la lista



current.previous.next = current.next
current.next.previous = current.previous

Resumen

- Una lista enlazada consiste en un objeto linkedList y un número de objetos Link.
- El objeto linkedList normalmente tiene una referencia al primer Link de la lista (first).
- Cada Link contiene datos y una referencia al siguiente Link (next).
- Las Listas Doblemente Terminadas poseen un puntero al último Link de la lista (last).
- Una Lista Enlazada ordenada, posee sus elementos en orden ascendente o descendente. Su tiempo de inserción es O(N) y eliminación O(N).
- Una Lista Doblemente Enlazada cada Link posee una referencia al su predecesor (previous) y sucesor (next). Permite recorridos en ambos sentidos.
- Un Iterador es una referencia encapsulada en un objeto que apunta a un Link de una Lista asociada. Permite recorrer la lista y ejecutar ciertas operaciones sobre los elementos de esta.

Experimentos

- Utilice el Applet LinkList para ejecutar operaciones de inserción, búsqueda y borrado, con lista desordenadas y ordenadas.
  Verificar si existe alguna ventaja en las listas ordenadas.

- Modifique el programa linkList.java de modo que ingrese ítems hasta que la memoria del equipo se ocupe casi a su máximo. Mostrar un mensaje después de cada 1000 ítems insertados.
  Comparar los valores en cada máquina probada.

Experimentos

- Implemente una cola de prioridad basada en una lista enlazada ordenada, donde la operación “remove” debe eliminar el ítem con menor valor.

- Una lista circular es una lista enlazada donde el último link apunto al primer link. Hay muchas formas de implementar una lista circular, algunas veces a través de un puntero al “comienzo” de la lista. Sin embargo, esto le quita el carácter de circular a la lista.
  Implemente una clase para una lista circular donde no existe ni comienzo ni fin. El único acceso a la lista es la referencia “current”, el que puede apuntar a cualquiera de los links de la lista. Esta referencia se puede mover a través de la lista según sea necesario. La lista debe manejar inserción, búsqueda, eliminación, y despliegue. Use un método step() para mover “current” hacia el siguiente elemento de la lista.