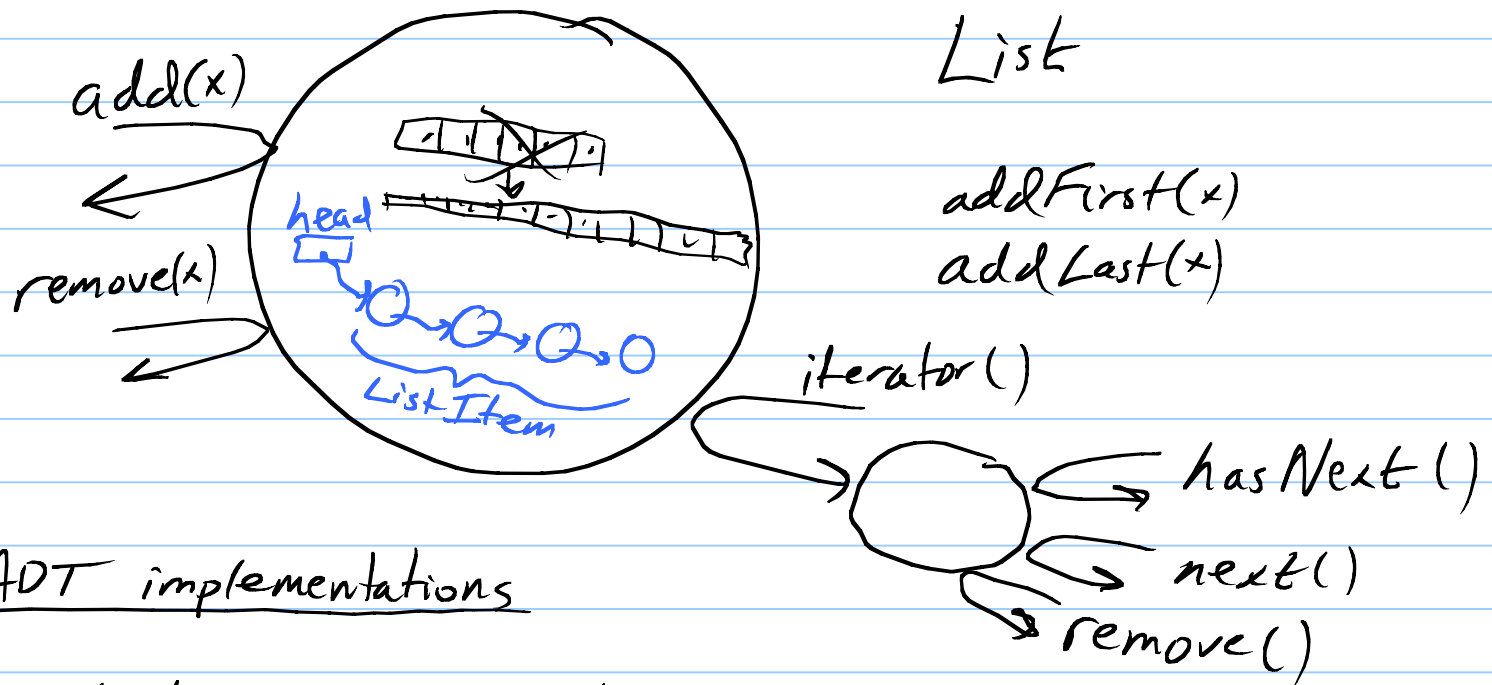


# Designing a List ADT implementation

Note Title

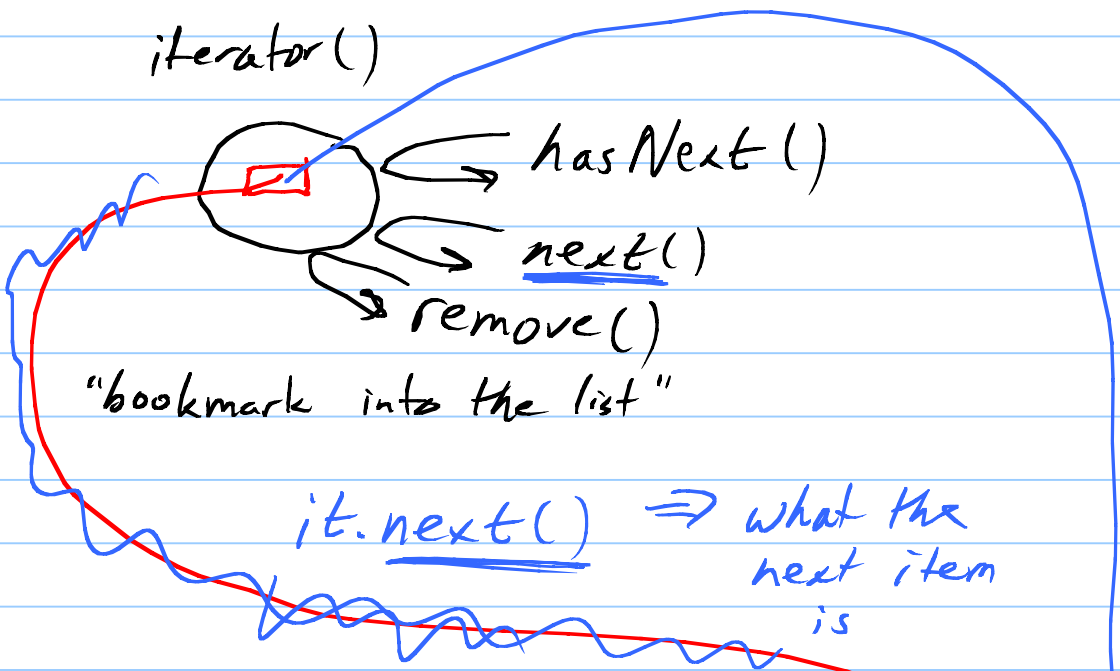
10/25/2007



## List ADT implementations

- **ArrayList** — encapsulates an array
- **LinkedList** — pointer-based implementation

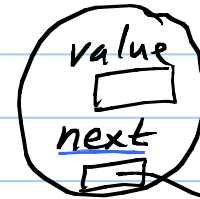
iterator()



"bookmark into the list"

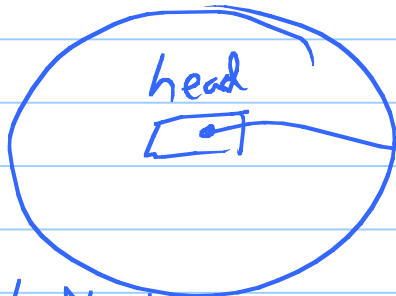
it.next() ⇒ what the next item is

ListItem

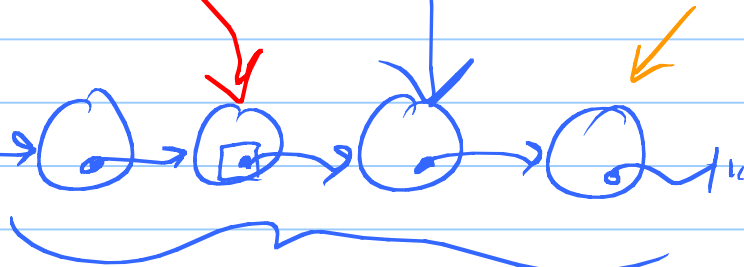


...  
another list item

listItem.next



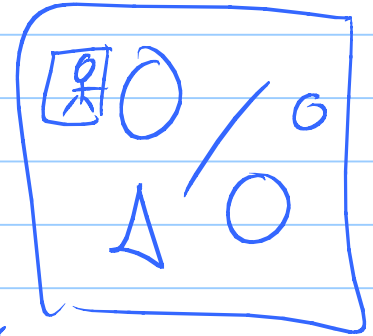
LinkedList



List Item

## Applications

- list of items on a GraphicsPanel
- list of documents waiting to be printed
- list of objects interested in timer notifications or mouse events



→ LinkedList

method name	parameters	effects	return
constructor	—	initialize an empty list	(empty list)
size	—	—	int — how big the list is
addLast	x	puts x at the end of the list	—
addFirst	x	puts x at the beginning of the list	—
remove	x	takes x out of the list	boolean — true if successful
contains	x	—	true if x is in the list
iterator	—	—	an iterator at the start

instance var.

tail ref.

head ref.

search alg.

class

Java's LinkedList ls

→ for (int x : ls) {

// can't remove from ls (don't have the  
iterator)

// Suppose you tried to remove this way!

if (x > 17)  
ls.remove(x)

}

ConcurrentModificationException