

Linked List Implementation

CSSE 221

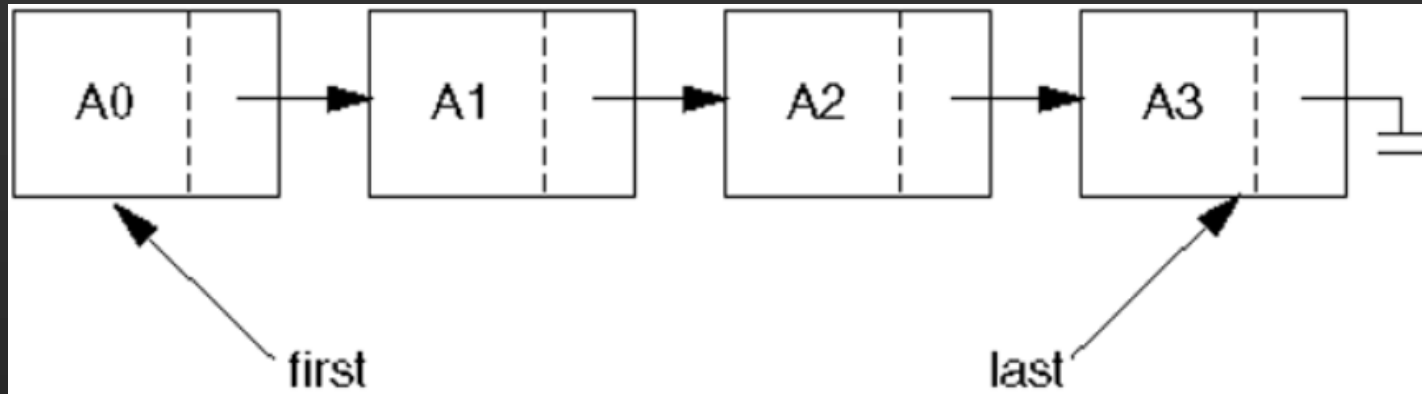
Fundamentals of Software Development Honors

Rose-Hulman Institute of Technology

Announcements

- Checkout the [SinglyLinkedList](#) project from your individual SVN repository.
- Final Exam Wednesday evening,
 - 6 pm to 10 pm, 0157 (Sec 1) & 0159 (Sec 2)
 - Start organizing your questions!

Linked List Basics

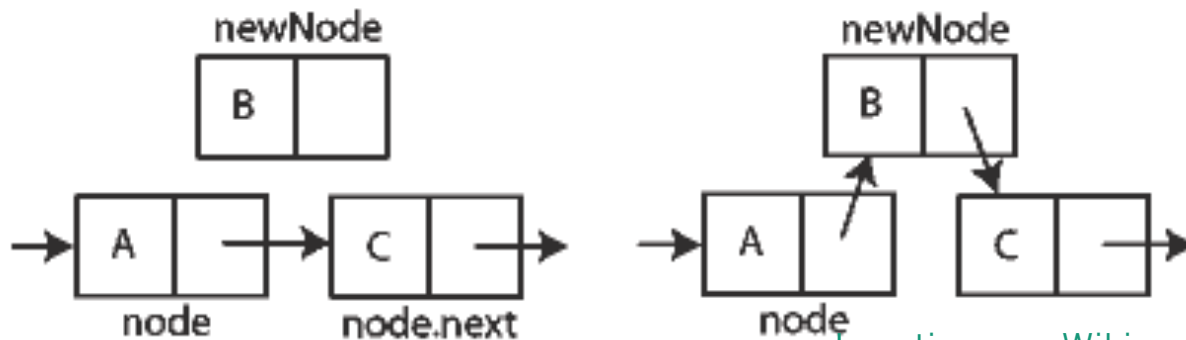


- Basic version:
 - The LinkedList class has only 2 fields:
 - a reference to the first node
 - a reference to the last node
 - Each node keeps track of the next node in the list.

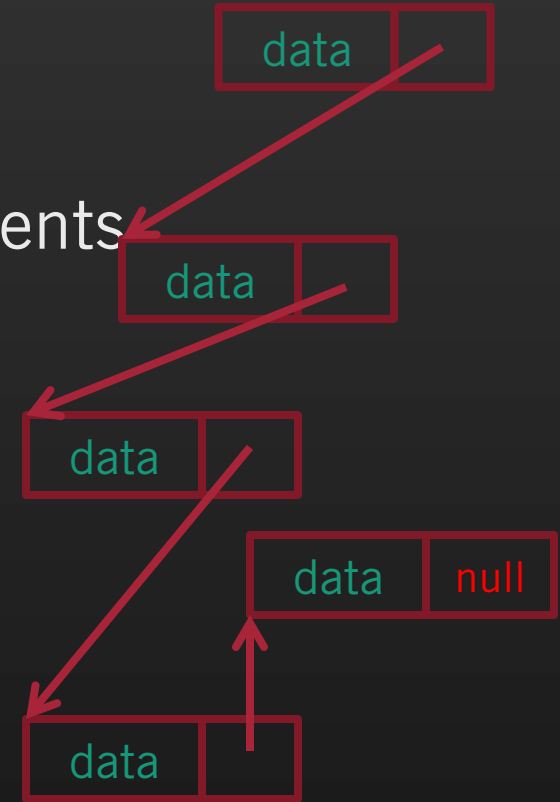
Another List Data Structure

- What if we have to add/remove data from a list frequently?
- **LinkedLists** support this:
 - Fast insertion and removal of elements
 - Once we know where they go
 - Slow access to arbitrary elements

“random access”



Insertion, per Wikipedia

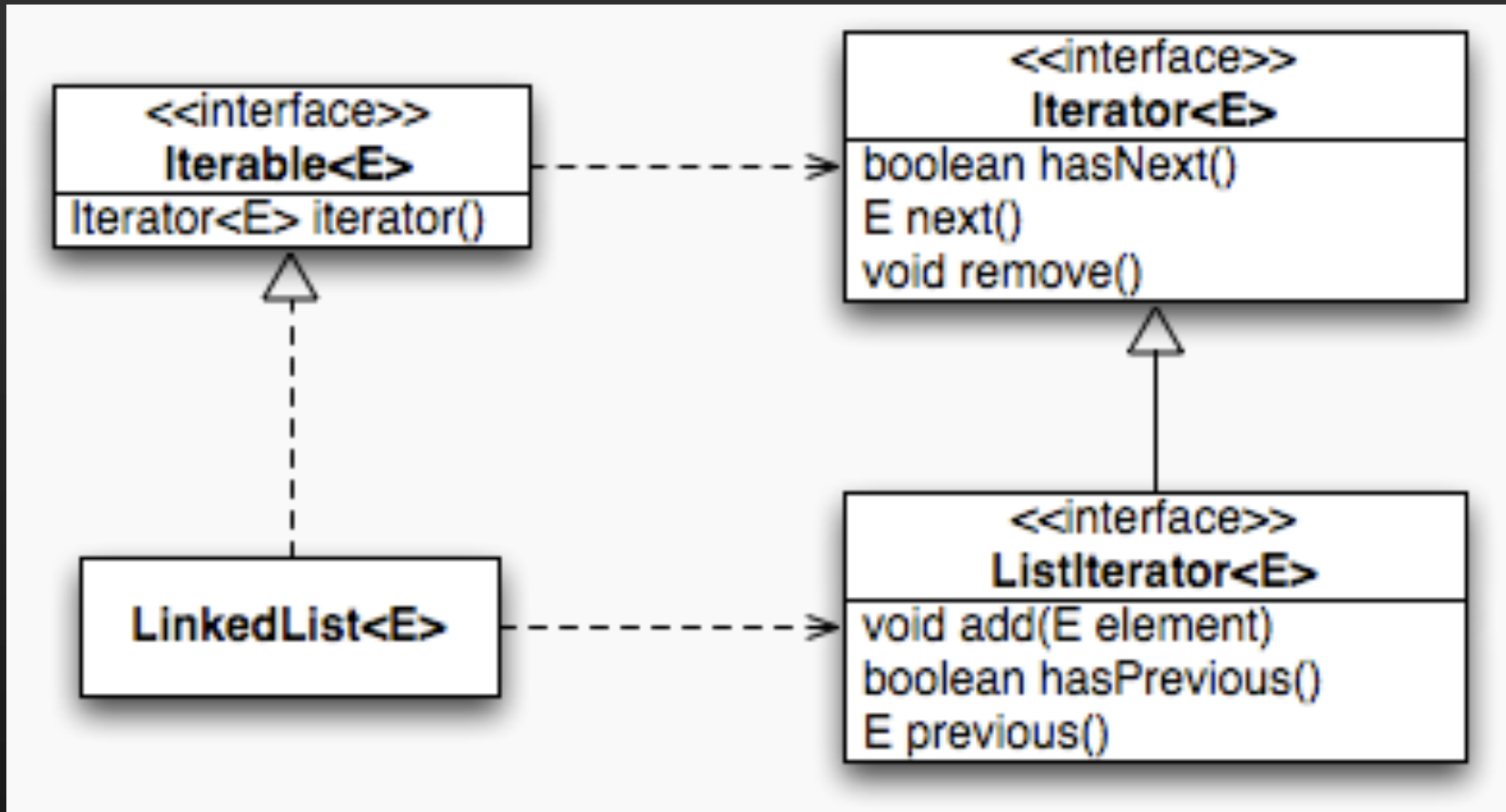


LinkedList<T> Methods

- `boolean add(T element)`
- `void add(int index, T element)`
- `int size()`
- `boolean contains(T element)`
- `boolean remove(T element)`
- `T get(int index)`
- `int indexOf(T element)`
- `T set(int index, T element)`

- What about traversing the linked list?
 - `LinkedList<T> implements Iterable<T>`

Traversing the LinkedList



Implementing LinkedList

- A simplified version, with just the essentials
- Won't implement all of the `java.util.List` interface
- Will have the usual linked list behavior
 - Fast insertion and removal of elements
 - Once we know where they go
 - Slow random access

Break

- <http://xkcd.com/379/>



Of course, the assert doesn't work.