Linked Lists + Exam Return

- ArrayList
List

- ArrayList

- List – ordered collection of values
  - List Interface
    - add(E elem)
    - add(int index, E elem)
    - contains(Observer o)
    - and many more!
List

- ArrayList

  - List – ordered collection of values

  - List Interface
    - add(E elem)
    - add(int index, E elem)
    - contains(Object o)
    - and many more!

- Array

  - Array – a data structure in which elements may be located by index
List

• ArrayList implements List
  • inherits all methods from List
    • ArrayList implements List’s methods using an array

ArrayList<String> array = new ArrayList<String>();

List<String> array = new ArrayList<String>();

List

• What are ArrayLists good for?
  • Constant time O(1)
    • size
    • isEmpty
    • get
    • set
    • add*
  * add is ~O(1)
  • Linear time O(N)
    • everything else
List

- list.add(“C”);
- list.add(“o”);
- list.add(“m”);
- list.add(“p”);
- list.add(“s”);
- list.add(“c”);
- list.add(“i”);

How would you
- add to the end, beginning, middle?
- remove an element?
Linked List

- LinkedList
  - List – ordered collection of values

  - List Interface
    - add(E elem)
    - add(int index, E elem)
    - contains(Object o)
    - an many more!

- LinkedList
  - different implementation of List from ArrayList
**Linked List**

- LinkedList
  - Nodes
    - data
    - pointer to the next node
  - Pointer to beginning and (sometimes) end

```
myHead

h -> e -> l -> o
```

- insert “l” after “e”

```
myHead

h -> e -> l -> o
```

```
myTail
```
Linked List

- insert “l” after “e”
  - find the location

![Diagram of a linked list with arrows pointing to nodes labeled 'h', 'e', 'l', and 'o'.]
Linked List

• insert “l” after “e”
  • find the location
  • create a new node

myHead -> h -> e -> l -> o -> myTail

Linked List

• insert “l” after “e”
  • find the location
  • create a new node
  • update newNode to point to ‘l’

myHead -> h -> e -> l -> o -> myTail

myNext
Linked List

- Insert “l” after “e”
  - Find the location
  - Create a new node
  - Update newNode to point to ‘l’
  - Update myNext to point to newNode

Linked List

- Singly linked list

- Doubly linked list
Linked List

- LinkedList vs. ArrayList

<table>
<thead>
<tr>
<th></th>
<th>get</th>
<th>add at end</th>
<th>add in middle</th>
</tr>
</thead>
<tbody>
<tr>
<td>LinkedList</td>
<td>O(N)</td>
<td>O(1)</td>
<td>Search time + O(1)</td>
</tr>
<tr>
<td>ArrayList</td>
<td>O(1)</td>
<td>O(1) but O(N) worst case</td>
<td>Shift elements</td>
</tr>
</tbody>
</table>

Bad Joke

- What did the ArrayList say to the LinkedList?
Next time

- Linked lists and code!
Exam

• Average – 71 ±11 points
• Median – 74.5 point
• Highest grade – 85 points

Exam

• Grade correction requests
  • Write ONLY the problem section and number on the front page
    • If there is a summation error, add “SUM” after the problem number
  • Hand you exam back to us BEFORE you leave class