Linked Lists

CS 62 - Fall 2015
Michael Bannister

FileIO

- File class:
  - represents a file or directory
  - doesn't have to exist
  - use the File.separator so that it doesn't matter what system we run on.
- Some methods that may be helpful:
  - delete()
  - exists()
  - createNewFile()
  - isFile()
  - isDirectory()
  - listFiles()
  - mkdir()
  - renameTo(...)

More FileIO

- Use the BufferedReader and PrintWriter classes for reading and writing to files.
- Have lots of useful methods
- PrintWriter out =
  new PrintWriter(new FileWriter(...));
- BufferedReader in =
  new BufferedReader(new FileReader(...));

Exceptions

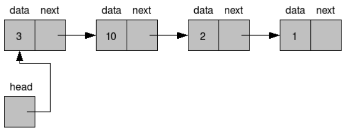
- Many methods/constructors throw exceptions
  - public String readLine() throws IOException
- Handle exceptions by try-catch construct
  - try {
    ... myFile.readLine() ...
    } catch (IOException ex) {
    code to be executed if exception raised
    }

Linked Lists

- Alternate implementation of lists
- Trade-offs in complexity
  - With ArrayList expensive to add at beginning of list
  - Linked lists inexpensive to add early
  - However, slow to access ith element.

Linked List

- Composed of Nodes
  - Think of as snap-lock beads
  - See code in structure5 library
    - From documentation page!
- See code in SinglyLinkedList
  - *Bailey - not std Java!*
  - keep track of head and size
  - Extends AbstractList -- look at on own!
    - Vector also extends AbstractList
- Also see SinglyLinkedListIterator



Linked List Algos

- Constructor
- addFirst, removeFirst
- get(i)
- indexOf(e)
- add(i,o)
- remove(e), remove(i)
- iterator

What is worst-case complexity of each?