Lecture 25: Even More Lists

CS 51G
Spring 2018
Kim Bruce

Announcements

- Exercise 13.3.4
- Quiz:
  - GUI and for loops
  - No recursion
- Recursion assignment

Structural Recursion Review

- Will always have (at least) 2 cases:
  - Base case: simplest possible structure
  - Recursive case: Have one or more instance variables with the same type as the object you are constructing. They hold “simpler” values of the same type.

Writing Recursion

- Write type of the object you are constructing
- Define an object or class representing the simplest possible object.
  - If it depends on parameters, it will be a class, otherwise it can be a simple object.
  - Methods should be trivial
- Define a class for the recursive case
  - Define complex object in terms of simpler pieces of the same type (and other objects as necessary)

Recursive Case

- Make sure construction terminates with base case at some point.
- Writing recursive methods
  - Assume methods work for all simpler cases
  - Write method using the methods on simpler cases
- Have faith!!

More Examples

- Scribbling again: *Done last time*
  - http://www.cs.pomona.edu/classes/cs051G/demos/ScribbleList/
  - http://www.cs.pomona.edu/classes/cs051G/demos/ScribbleCollection/
  - Scribble represented as a list of Lines, while scribbleCollection is a list of scribbles
    - essentially a list of lists!
  - Notice use of return, e.g., in contains for Scribble & lots of places in ScribbleCollection
  - Also, see how for loops iterate over lists!

List Operations

- See Documentation!
  - at, add, remove, contains, indexOf, ++, etc.
  - If aList has n elements then can write
    - aList.at (k) put (val) for k = 1, 2, ..., n, n+1 to update element in kth slot.
    - atList.at (n+1) put (val) is like add, as it extends list, while smaller values of k simply update the value in slot without extending.

Questions?