Lecture 26
Finishing up with Strings Design!

Strings

- Many programs manipulate text
  - Email
  - Web browsers
  - Peoplesoft
  - Spreadsheets

Link Extraction

- Let’s look at the html file for the CS 134 page
- Note the text mark-up commands:
  - `<a href=
  - `<table>`
- Now let’s write a method that will extract all the links from a file like this. Why?
  - Check that no links are dead
  - If we were a spammer, might want to look for email addresses
  - If we were Google crawling the web,...
  - If we were an organization looking for addresses for outreach
- The text in a file is just one giant String

private String findLinks(String fullPage) {
  // A lower case only version of the page for searching
  fullPage = fullPage.toLowerCase();
  // The A tags found so far
  String result = "";
  // Start of <a tag specification
  int tagStart = fullPage.indexOf("<a", 0);
  while (tagStart > -1) {
    int tagEnd = fullPage.indexOf(">", tagStart);
    String tag = fullPage.substring(tagStart, tagEnd + 1);
    result = result + tag + "\n";
    tagStart = fullPage.indexOf("<a", tagEnd);
  }
  return result;
}

Search and Replace

- Note that concatenation (+) and substring(...) together give us the ability to break apart and build new Strings.
  - Kind of like “paste” and “cut”!

private String searchAndReplace(String search, String replace, String text) {
  // find first occurrence of search string
  int pos = text.indexOf(search);
  while (pos > -1) {
    text = text.substring(0, pos) + replace +
    text.substring(pos + search.length());
    // start from end of replacement (in case replacement
    // contains the search string...
    pos = text.indexOf(search, pos + replace.length());
  }
  return text;
}
Miscellaneous

• Note that our String demos extend Controller
  – No canvas needed for these
• JTextArea
  – A GUI component very similar to a JTextField, but that allows more than a single line.

Object Oriented Design

Steps to follow in doing design

1. Determine Classes:
   – What are the major components/objects of the program?
2. List the properties and behaviors of each major type of object (i.e., each class)
   – Just in words for now (can write them as comments)
3. Model the properties as instance variables
4. Model the behaviors with methods
   – Think hard about parameters
5. Begin to fill in the details
   – Might have to iterate back to Step 4*

It's great when the plan is perfect from the start, but it is fine to go back to earlier steps if a later detail makes you realize you need a change.