CSC207.01 2013F, Class 53: Using Java from the Command Line

Overview

- Preliminaries.
    - Admin.
    - Questions/comments on the homework.
- A brief overview.
- Ant.
- Lab.

Preliminaries

Admin

- Candy Canes and Clementines!
- Study break 8-10 tonight, Smoothies! (May run out early.)
- Who has an OCS recommendation that I need to fill out?
    - EW, DG, JGS, ...
- I hope to distribute most of the exam tomorrow. (I’ll probably leave the code until HW11 is turned in.)
- Upcoming extra credit opportunities:
    - CS Extra: Multiple Models of Mediascripting
    - CS Table Friday: TBD

Questions/Comments on the Homework

Sam’s short comments

- You are effectively building a tree.
- If you’re building a tree-like structure, do you want to use iteration or recursion?
- Basic operation: Look at the next character
    - Open brace: It’s an object, parse the object
    - Open square bracket: It’s a list
    - Double quotation mark: It’s a string (don’t have to worry about \u, but do have to worry about \" and \)
    - Digit: It’s a number
    - Etc.

Should we split at quotation marks or simply step through the characters?

You should step through the characters. Quotation marks can have too many meanings.

When is a digit going to happen?

{"id":23}

A brief overview

- Goal: Escape from Eclipse
- Why?
  - May be more efficient with other editors.
  - Want less startup time.
  - Want to know what’s going on "behind the scenes"
- `javac` - Compile a .java file
- `java` - Run a compiled .java file
- `CLASSPATH` - Environment variable that tells where to look for other compiled Java files.
  - In most Unix/Linux variants, there are user-specific variables that let you configure the system (e.g., where to look for programs, where to look for Java stuff, what editor to use by default, ...)
  - In bash, we set them with $ export VARIABLE=value; view with $ printenv VARIABLE
- `jar` - Join lots of compiled Java files into a single "package"
- `javadoc` - Generate documentation
- `junit` - Run unit tests

Ant

- A typical project has lots of things to do - compile, join into an application, test, run a program, clean up temporary files, etc.
- In C, we write Makefiles.
- The folks at Apache built Ant as a Java-oriented alternative to Make.

Lab