CSC207.01 2013F, Class 07: Arrays in Java

Overview

- Prelim
  - Admin.
  - About HW 3
  - About HW 2
- ADT Design
- Arrays

Admin

- A final reminder to register for the 2nd-year retreat. It looks like a good opportunity to do lots of things - talk more about group work, learn from alums, think about choosing a major, etc.
  - Email [greggjol] to get the links
- It sounds like there was some demand that I focus on the syntax and semantics of imperative Java, so I've rearranged the syllabus (including today) to go over it.
  - Rearranging stuff put me even further behind! Readings may not be ready in time, and class sessions will probably focus on lecture + discussion + recitation.
  - And my class outlines may not be as deep as they usually are.
- Reminder: Mentor session Wednesday night at 8pm
- Readings for Wednesday (expect them later today):
  - Strings in Java
  - Numbers in Java
- Homework 3 is ready!
- EC opportunities:
  - Humanities Center Speaker Sarah Hendron, Wednesday, 7:30 p.m., JRC101 Waking the Machines: Art, Design, and Adaptive Technology
  - Learning from Alumni 2:15-4:05 Thursday: Ian Lunderskov '08
  - Tentative Thursday Extra: Matt Atherton '95
  - CS Table, Friday: Trusting Trust.
  - More?

About HW 3

See the homework. Lots of chances for extra credit.

About HW 2

- Many 4-5 hours
- Lots of problems with Eclipse and Git(Hub)
  - Pull requests are a pain
    - So add an administrator (easier for you than for me)
  - Pulling changes was hard
  - Debugger is hard
    - Write experiments, rather than unit tests, when you want to debug
  - What’s the workspace?
    - A default place to store stuff, including config (I believe)
    - I’d recommend ~/workspace
    - But your projects generally won’t go in the workspace, maybe put those in ~/CSC207
- Getting the syntax right in parts C and D took a lot of time
- Why the average problem?
  - Think about extreme cases!
  - A real issue

Java Things to Discuss

- Java is object oriented: We build and use objects
- Sometimes, we just want to be imperative and write methods - no objects really needed
  - Methods not associated with objects are called "static methods"
  - Until you build objects, everything should probably build static
- Related issue: Classes serve three basic purposes
  - Collections of static utility methods - "Utility class"
  - A repository for the main method - "Main class"
  - Templates for objects - "Template class"

ADT Design

- Sam can’t speak today
- An ADT is a set of operations and values that have some common purpose
  - An array has the purpose of storing values indexed by integers
  - A list has the purpose of storing values that we can iterate through
  - A PrintWriter has the purpose of generating output
- We generally achieve that purpose through the methods
  - Arrays need methods to get the value at position i, set the value at position i, and get the length
  - PrintWriters need methods to print things and to print newlines
- We also look at the practical applications of such ADTs

Once we’ve done all that, we look for ways to implement the ADT

Design confilct: minimalist vs. maximalist * Scheme lists:
* Minimalist: cons, car, cdr, null, null? * Maximalist: member, trim, split, index-of, element-at, map, caaddar, length ...

Arrays

- Declare

  ```
  TYPE[] NAME
  ```

- Create

  ```
  new TYPE[SIZE]
  new TYPE[] { VAL1, VAL2, VAL3, ... }
  ```

- Access values

  ```
  NAME[INDEX]
  ```

- Set values

  ```
  NAME[INDEX] = VAL
  ```

- Get length

  ```
  NAME.length
  ```