Topics: 2-d arrays, semester wrap-up

Reading (JV): Sec 6.3, review char and String in Sec 2.4, 2.6

Accessing a 2-d array

Given a reference x that points to a 2-d integer array...

- What is its height (# of rows)?
- What is \[ x[0] \]?
- What is the length of the first row?
- How to access last element in row 2?
- How to access last element in last row?

Example 1: 2-d array and a useful pattern

Given a 2-d integer array \( x \), find the maximum value in the array. Assume the array is rectangular.

What if the array is ragged instead of rectangular? Suppose all rows exist but the rows have different lengths.

What if not all rows exist and the existing rows have different lengths?

Example 2: Strings in a 2-d array

Given seat, a 2-d array of Strings that stores a seating plan, complete the program segment below to find the row and seat number of the person whose name is given through user input. Array seat has dimensions just big enough to store the entire seating plan including internal spaces. Assume all rows have length > 0.

```java
String target = Keyboard.readString();
```
String methods

Below are some useful methods of the String class. Let s refer to the String “CS100M” in the examples below.

<table>
<thead>
<tr>
<th>Expression</th>
<th>Returned value</th>
</tr>
</thead>
<tbody>
<tr>
<td>s.length()</td>
<td></td>
</tr>
<tr>
<td>s.charAt(0)</td>
<td></td>
</tr>
<tr>
<td>s.indexOf('0')</td>
<td></td>
</tr>
<tr>
<td>s.equals(&quot;CS100&quot;)</td>
<td></td>
</tr>
<tr>
<td>s.toLowerCase()</td>
<td></td>
</tr>
<tr>
<td>s.toUpperCase()</td>
<td></td>
</tr>
</tbody>
</table>

More String methods are listed in Figure 2.8 in Section 2.6 of the text book. Appendix M in the text book is a good reference to the Java Class Library. Pages 886-889 cover the String class. Use these listings as a resource, but don’t go memorizing all the methods!

What we learned this semester...

- Develop/implement algorithms for problems
- Develop programming skills
  - Design, test, debug, document, demonstrate
- Apply programming languages
  - Control structures
  - Function/methods for reducing redundancy
  - Data structure
  - Fundamentals of object oriented programming
- Specific tasks
  - Sorting
  - Simulation of systems
  - Text and string processing
  - Handling input/output files
  - Plotting numeric data