Announcements

- **Section** in computer labs this week
- P6 due Thurs, 5/5, at 6pm
- Final exam: 5/13, noon-2:30pm
- Let us know now if you will have an exam conflict. Email Kelly Patwell with your exam info.

Previous Lecture:

- Polymorphism
- Object class
- Abstract

Today’s Lecture:

- 2-d array
- Reading: Sec 7.6

2-d arrays

- A 1-d array is a *list* of values (references)
- A 2-d array is a *table* of values (references)
- 2-d array is referenced using two index values
- 2-d array in Java is really a 1-d array of 1-d arrays (i.e., an array of objects)
  - Orientation (row, column) is how we choose to visualize the “table”
  - Convention: row-major

Multi-dimensional array

- Can have as many dimensions as you want
- A 2-d array is a 1-d array of 1-d arrays. Each 1-d array has its own constant *length* ⇒ you can have a ragged 2-d array.

Creating a 2-d array

1. Declare a reference `x` for a 2-d integer array
2. Create a 2-by-3 integer array `y`
3. Create the following array:
```
  2 4 6
  8 1 3
```

Creating and accessing 2-d array

- Declare a reference `x` for a 2-d integer array
  ```java
  int[][] x;
  
  Create a 2-by-3 integer array `y`
  ```java
  int[][] y = new int[2][3];
  ```
- Create the following array:
  ```java
  2 4 6
  8 1 3
  ```
- Given reference `x` that points to a 2-d integer array, ...
  - What is its height (# of rows)? `x.length`
  - What is `x[0]`? Reference to 1st row
  - What is length of 1st row? `x[0].length`
  - How to access last element in row 2? `x[1][x[1].length-1]`
  - How to access last element in last row? `x[x.length-1][x[x.length-1].length-1]`
Accessing a 2-d array

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

1. What is its height (# of rows)?
2. What is `x[0]`?
3. What is the length of the first row?
4. How to access last element in row 2?
5. How to access last element in last row?

What if . . .

- The array is ragged instead of rectangular? Suppose all rows exist but the rows have different lengths.
- Not all rows exist and the existing rows have different lengths?

Example 1

Given a 2-d integer array `x`, calculate the sum of all entries in the array. Assume the array is rectangular.

```java
// Given 2-d array referenced by x
// calculate sum of all elements
int sum = 0; // sum so far
for (int r=0; r<x.length; r++)
  for (int c=0; c<x[r].length; c++)
    sum += x[r][c];
```

Example 2

Given a 2-d array `m`, re-order the rows such that the row with the highest row sum is the first row.

```java
//calculate row sums
//find index of row with max sum
//swap row of max sum with row 0
```