

Introduction to Comp. Sci., Homework 5

Due at 12pm on Monday, March 11

Readings from Liang

Read chapter 8, “Objects and Classes,” in Liang. Don’t worry if you don’t get all the details.

Optional readings and exercises from HFJ

Read chapters 2 and 3 of Head First Java. Do the exercises, and check your answers against those provided at the ends of the chapters.

To be turned in

Create a Java class, `hw5.Matrix`, that represents an $r \times c$ matrix (r rows, c columns) with a single array of `double`. It should have these public methods:

- A constructor that takes two `int` arguments, `rows` and `columns`. It should initialize a non-public member array of `double` to be `rows * columns` long. Each element of this array will correspond to a single entry in the matrix. It should save the `rows` and `columns` arguments so they can be returned later.
- `numRows` and `numColumns` should each take no arguments and return the number of rows/columns in the matrix.
- `get` should take two `int` arguments, `row` and `column`, and return the `double` in the specified position in the matrix, where rows and columns start being numbered at zero. If the requested row or column is negative or greater than or equal to the number of rows/columns, signal an error by throwing an `IllegalArgumentException`. You can do that with this statement:

```
throw new IllegalArgumentException();
```

If a caller tries to `get` the value of an element without calling `set` on it first, `get` should return zero. Since zero is the default for elements of `double` arrays, this should require no extra work.

- `set` should take two `int` arguments, `row` and `column`, and one `double` argument, `v`. It should set the specified element of the matrix to `v`. If the row or column is invalid, it should signal an error as with `get`.

Here's some example code that uses `Matrix`:

```
// Create a 5 by 6 matrix.
hw5.Matrix m = new hw5.Matrix(5, 6);
// Set element 2, 4 to 3.0.
m.set(2, 4, 3.0);
// Print out that element; should print "3.0".
System.out.println(m.get(2, 4));
```

I recommend laying out the elements of the matrix in the array columnwise, using `%` and `/` to figure out which element of the array corresponds to which row and column. Here is how the elements of an array with three rows and four columns could be laid out:

```
0 3 6 9
1 4 7 10
2 5 8 11
```

I wrote a smoke test for your programs. You can run it with (Mac):

```
$ testing/test.sh Hw5Test
```

or (Windows):

```
> testing\test.ps1 Hw5Test
```

Remember to run `svn update` to synchronize my tests into your working copy.