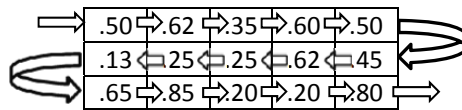


CastleDefense

The purpose of this assignment is to practice:

- using the ArrayList class
- reading from a file
- working with 2-D arrays
- testing your code
- *EDITING AND REVISING YOUR WORK*

A defense scanner exists that scans a rectangular area of a castle's defenses and collects data into a 1-dimensional array. Each data value scanned is a number representing the strength of the defenses in that position on the castle. The scanner scans back and forth across the castle face, alternating between left to right and right to left, in the pattern indicated below by the arrows.



The scanner records data into a 1-dimensional ArrayList of double values. This 1-dimensional array of data is transferred into a 2-dimensional array of doubles, which reconstructs the original view of the defenses on the face of the castle.

You are writing a program that detects weak spots in a castle's defenses. Call your program CastleDefense.java. Your program should include the following:

1. A main method that
 - a. Reads doubles from a file called data.in and store in an ArrayList.
 - b. Calculates the number of rows and columns needed to store the data. This should be as close to a square as possible – in other words, if there are 12 values, then you should create a 3 x 4 grid, rather than a 2 x 6 or 12 x 1 grid. Create a CastleDefense object using this data.
 - c. Print the grid and the upper left coordinates of the weakest 2 X 2 sector to the monitor.
2. A CastleDefense class definition that includes
 - a. A private field for the grid, declared as follows:
`private double [][] grid;`
 - b. A constructor with the following header:
`public CastleDefense(int numRows, int numCols, double[] scan)`
The constructor allocates and initializes the grid by transferring the 1-dimensional array of data into a 2-dimensional array of doubles.
 - c. A method with the following header:
`public double getAverage(int startRow, int endRow, int startCol, int endCol)`
This method calculates and returns the *average* strength of the castle defense in the area specified by the parameters, inclusive.
 - d. A toString method that outputs the grid.
 - e. A method with the following header:
`public void findWeakest()`
This method should repeatedly calls the getAverage method, finding the weakest 2 X 2 area on the castle face. The starting row and starting column of this area should be output. You can assume that your grid will have at least 2 rows and 2 columns.

Sample run

The input file containing the following:

.5 .62 .35 .6 .5 .45 .62 .25 .25 .13 .65 .85 .20 .20 .8

Should produce the following output:

run:

0.5	0.62	0.35	0.6	0.5
0.13	0.25	0.25	0.62	0.45
0.65	0.85	0.2	0.2	0.8

The weakest 2 X 2 on the castle begins at (1, 2) with an average strength of 0.3175

BUILD SUCCESSFUL (total time: 0 seconds)