[UAB](http://main.uab.edu/) -> [ENG](http://www.eng.uab.edu/) -> [ [ECE](http://www.uab.edu/engineering/departments-research/ece) / [ECE Intranet](http://www-ece.eng.uab.edu/) ] -> [ [DGreen](http://www-ece.eng.uab.edu/DGreen/redirect/dgreen-internet.html) / [DGreen Intranet](http://www-ece.eng.uab.edu/DGreen/) ] -> P2

**EE333 Programmette 2 - Fall 2011**

**Task - IndexTable Class**

Your task is to design and build a class called IndexTable. This class will model a indexing table in several ways in several ways. Each indexing table shall have a unique serial number.

**Background on Indexing Table**

An indexing table is used to move items around such as a slide under a microscope or a moving a product under a tool for a manufacturing operation. The table itself is capable of moving to many discrete positions based on integer numbers.

In P2, you are to model an indexing table which features several modes of positioning. The model shall keep track of where it is as well as its error state. One way to be in an error state is to receive a command that can not be executed.

**IndexTable Characteristics**

* Construct: maximum values of the x and y coordinates are input (serial number applied with constructor) When constructed, the index table shall be at position 0, 0 and be in a non-error state.
* Queries:
  + isOK() - returns true if not in error state, false otherwise
  + isError() - logical complement of isOK()
  + isHome() - returns true if table is known to be at 0,0 else returns false
  + getX() - returns present x coordinate of table
  + getY() - returns present y coordinate of table
  + toString() - “<serialNumber>, <status>, <xPosition> <yPosition>”

Don’t show the < or > in toString example

* Commands:
  + resetError() - clear error condition without moving table (if possible)
  + reset() - clear error, move table to home position
  + move(int x, int y) - move index table to x,y
  + moveRelative(int dx, int dy) - move index table to new position by specified incremental amount
  + moveVector(int r, int theta) - move index table from present location to new location in direction of vector. theta == 0 means along the the x axis and theta == 90 means along the y axis. Theta can be any value. The final position should be rounded to nearest index position.

**TestIndexTable**

Design and build a class that will have a public runTest() method which will run several tests to exercise the IndexTable class. The tests should exercise each query and command. Remember, tests should both exercise functionality and grade the response.

**Obey Java Documentation Style**

Continue to use the specified [documentation standard for Java source code](http://www-ece.eng.uab.edu/DGreen/java_ex/JavaDocStyle.html).

**Delivery**

You shall produce source code that complies with the documentation standards. Your program MUST show your name and BlazerID near the top of the listing and display out your name and BlazerID at the start of the test run. Produce a blazerid-p2.zip file containing a directory blazerid-p2.zip containing all files (at least .java and .class files) and submit it using the assignment tool of Blackboard Learn by class time of the due date (or if you attend class and sign the class roll for the day, you can automatically have an extension to 11:59pm of the same day.)

The programmette will be graded by this [rubric](http://www-ece.eng.uab.edu/DGreen/ee333/ProgrammetteGrading.png).

Last modified: 30 August 2011

David Green [.](http://validator.w3.org/check?uri=referer)