[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/) ] -> [EE333](http://www-ece.eng.uab.edu/DGreen/ee333/) -> P3

**EE333 Programmette 3 - Fall 2011**

**Task - Modified Index Table**

Your task is to modify the IndexTable as described below, add a text user interface to allow a user to use the IndexTable, and to create JUnit tests to test the new IndexTable model.

We are going to (weakly) incorporate the concept of time. It will take time for the table to move. We could do this in several ways:

1. Put in time delays on all move methods to accommodate the movement
2. Put in a status call like “is\_complete()” which would return true or false enough time had gone by
3. Put in a status call like is\_complete() which always returns true but also sets up the value for “time\_elapsed()” method which returns the total time elapsed since the last invocation of time\_elapsed().
4. Create a clock concept that allows time to move at simulation speed.

For P3, we will choose option 3.

Add a boolean is\_complete() which always returns true but updates (incrementally) an internal elapsed time to reflect the time of the last movement.

Add a double time\_elapsed() method which returns the time elapsed by the table movement assuming that the table moves linearly

* 5 units/second if distance less than or equal to 20 units
* 50 units/second if distance over 20 units

Create a text user interface (TUI) which controls the table from the command line.

An example log of running your program:

java IndexTableDemo

IndexTableDemo 1.0 by yourname

Command:createtable 1000 1000

IndexTable status: 1, OK, 0, 0

Command:createtable 500 500

IndexTable status: 2, OK, 0, 0

Command:move 1 100 100

Command:iscomplete?

true

Command:moverelative 1 10 -10

Command:iscomplete?

true

Command:tablestatus 1

IndexTable status: 1, OK, 110, 90

Elapsed time: 4.0 seconds

Command:movevector 2 100 45

Command:iscomplete?

true

Command:tablestatus 2

IndexTable status: 2, 71, 71

Elapsed time: 2.0 seconds

Command:reset 1

IndexTable status: 1, OK, 0, 0

Command:junk with stuff on line

Command:quit

java IndexTableDemo -h

IndexTableDemo 1.0 by yourname

This program is supplied for P3 of EE333 Fall 2011.

Commands:

createtable {max\_x} {max\_y}

move {table\_sn} {x} {y}

moverelative {table\_sn} {dx} {dy}

movevector {table\_sn} {r} {degrees}

reset {table\_sn}

time\_elapsed {table\_sn}

quit

Lines that start with any other words are ignored.

Lines that are missing required data are ignored.

Ten IndexTables can be created.

Commands to IndexTables that don't exist are ignored.

Commands with improper arguments are ignored.

Note that table\_sn should also correspond to the creation order.

Built with NetBeans.

In implementing the above, respect the modeling. As the models for information, don’t store information in the IndexTableDemo application that you have stored in the models (except temporarily). You must support at least 10 IndexTables. If you support more than 10 IndexTables, then adjust the help message appropriately. Consider using NetBeans for this assignment. If you do not, modify the help message appropriately.

Be sure to test the precise case above and include it as a sample run.

**JUnit Tests**

Create a JUnit test class that tests the new IndexTable model.

**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. Produce a blazerid-p3.zip file containing a directory blazerid-p3 of all files (at least .java and .class files) and submit it using the assignment tool of Blackboard Vista 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: 13 September 2011

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