Mini Bonus HW1: Train Your Turtle to Draw on Command (20pts)

DUE DATE: Friday March 7th 7pm with 5 hour grace period

Hopefully by this point, you have had already some experience with file I/O (file input/output) and drawing with turtles. Now you will be required to write a program that takes in a text file of commands and translate those commands into turtle movements on a world.

Analyzing the Text File Format

You may assume the format of the text file will always be the type of the command first and then the parameters that command takes in. The commands you will be expected to handle are:

Command type	Parameters	What it does
world	width, height	Creates the world based on parameters
forward	distance	Moves the turtle forward based on parameter
turn	degree	Turns the turtle based on parameter
moveTo	x coordinate, y coordinate	Moves the turtle to a specific coordinate

TurtleDrawing

In the `TurtleDrawing` class, you will be required to write one new constructor and three new methods: `extractCommands`, `drawCommands` and `toString`.

public TurtleDrawing(String turtleCommandsTextFile)

The constructor will take in the file path of the turtle commands text file. First initialize the commands `LinkedList` to be an empty `LinkedList` of `TurtleCommand` objects (meaning you must use `<TurtleCommand>`). Then call the `extractCommands` method passing in the `turtleCommandsTextFile`.

public void extractCommands(String turtleCommandsTextFile)

In this method you will need to use a `BufferedReader`, `Scanner` or similar class to actually read the data from the text file. Go through the text file and extract each line at a time. Assume the first line of the text file will always be a world command. You can assume that each line will contain only one command. The first item in the line will be the command type and followed by its parameters. You can also assume that each line is correct meaning that each command has the exact number of parameters it needs (as outline in the previous table).

For each line, create a new `TurtleCommand` placing the command type and its parameters in that `TurtleCommand`. Then add that `TurtleCommand` to the big `LinkedList` of `TurtleCommand`s called `commands`. Do this until you have extracted all of the commands.

public void drawCommands()

This is the method where you draw all of the commands in your `commands` list. Loop through your entire `commands` list and check the type of each `TurtleCommand`. If the command type is `"world"` then you have to create the world based on the parameters in that `TurtleCommand`, add a new `Turtle` to that world and show the world. If the command type is `"forward"` then call the `forward` method on the turtle you created before. You will do something similar with the `"moveTo"` and the `"turn"` command types. Remember to call the provided `pause` method after each command so you can see the progression of the commands.

Command type	What to do
world	Create a new world based on parameters Create a new turtle and put it on that world Show the world*
forward	Call the `forward` method based on parameters on the previously defined turtle*
turn	Call the `turn` method based on parameters on the previously defined turtle*
moveTo	Call the `moveTo` method based on parameters on the previously defined turtle*
	* Remember to `pause` after.

Main method

In your main method create a new `TurtleDrawing` and give it the path to the turtleCommands text file. Then call the `drawCommands` method and print out the `toString` of the `TurtleDrawing`.

What to Turn In

- `TurtleDrawing.java`

Where to Turn In

- T-square