COSC 1P02 Assignment 2

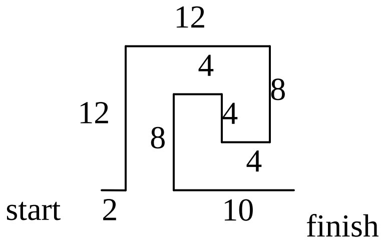
"It's all Greek to me"

***Due: Oct. 21, 2016 @ 4:00 pm** (late date Oct. 24 @ 4:00 pm)*

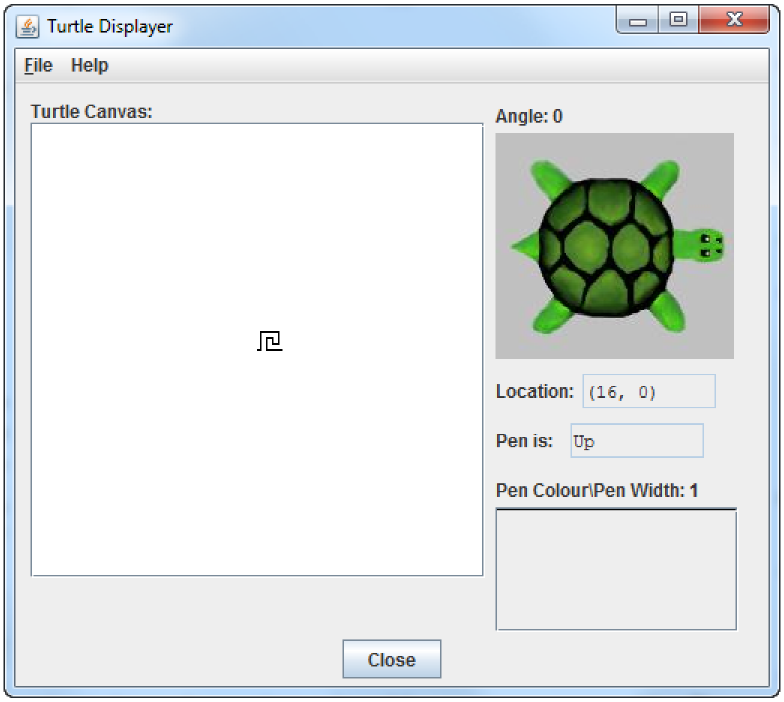
The emphasis for this assignment is methods without parameters. In preparation for this assignment, create a folder called `Assign_2` and three subfolders `Assign_2_A`, `Assign_2_B` and `Assign_2_C` for the DrJava projects for the assignment.

Part A

A Greek Key is a border pattern commonly seen in classical Greek and Roman art and architecture. It is an example of a general kind of pattern called a meander (see: http://en.wikipedia.org/wiki/Meander_%28art%29). Although there is an indefinite number of possible patterns, we are going to build one using an order 3 key pattern that looks like the following:



As part of a package `Assign_1_A`, write a Java class `GreekKey` using Turtle Graphics to draw the above Greek Key pattern, with line lengths as indicated, starting at the center of the display. That is, the first line is drawn from the center to the right as below:



Location: (16, 0)

Pen is: Up

Pen Colour\Pen Width: 1

Your solution should include the method:

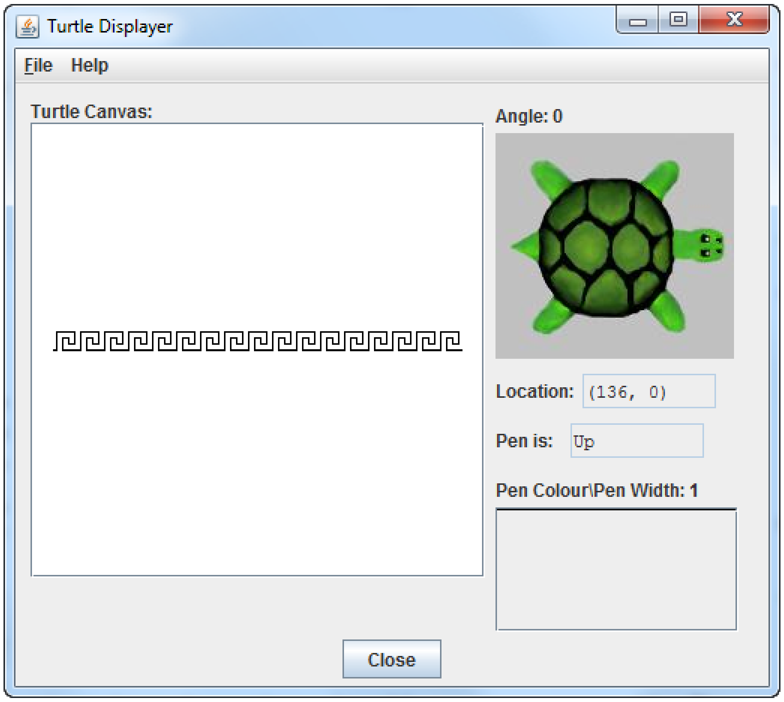
```
    private void drawKey ( ) {
```

that draws the Greek Key pattern from the turtle position forward and to the left.

revised: 06/10/2016

Part B

Building on the Greek Key pattern from Part A and as part of the package `Assign_1_B` write a Java class `GreekRow` that draws a repeating Greek Key pattern (a row with 17 Greek Keys). The row should be centered and drawn left to right on the canvas as shown below:



Angle: 0

Location: (136, 0)

Pen is: Up

Pen Colour\Pen Width: 1

There should be no space between the keys. That is the second key begins exactly where the first ends.

Your solution should include the method:

```
private void drawRow ( ) {
```

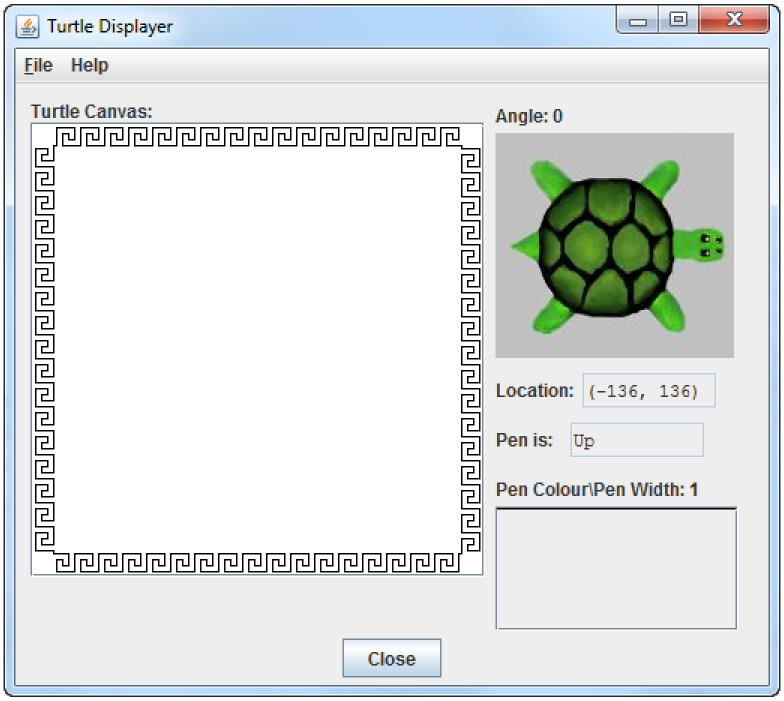
that uses the method `drawKey` to draw a row of Greek Key patterns from the turtle position in the direction of the turtle.

Hints:

- Copy and paste the `GreekKey` class from Part A as a starting point for the `GreekRow` class and then extend the code appropriately.
- Since there are a lot of lines, use a `FAST` turtle. You can specify a speed for the turtle in the creation expression such as:
  ```
  yertle = new Turtle(FAST);
  ```
  The constant `FAST` can be imported from the `Turtle` class via:
  ```
  import static Media.Turtle.*;
  ```

Part C

Building on the row of Greek Keys from Part B, and as part of the package `Assign_1_C`, write a Java class `GreekBorder` that draws a border using a repeated Greek Key rows. The border should be centered on the canvas as shown below:



Angle: 0

Location: (-136, 136)

Pen is: Up

Pen Colour\Pen Width: 1

The sides should all be joined (i.e. the second set of patterns begins at the same point as the first set of patterns ends).

Your solution should include the method:

```
    private void drawBorder ( ) {
```

that uses the method `drawRow` to draw four rows of Greek Key patterns clockwise from the turtle position starting in the direction of the turtle.

Hints:

- Copy and paste the `GreekRow` class from Part B as a starting point for the `GreekBorder` class and then extend the code appropriately.
- Since there are a lot of lines, use a `FAST` turtle.

Submission:

Details regarding preparation and submission of assignments in COSC 1P02 are found on the COSC 1P02 Sakai Site as `Assignment Guidelines` under `Course Documents`. This document includes a discussion of assignment preparation, programming standards, evaluation criteria and academic conduct (including styles for citation) in addition to the detailed assignment submission process copied below.

To prepare and submit the assignment electronically, follow the procedure below:

1. Ensure your folder (`Assign_2`) for the assignment has subfolders `Assign_2_A`, `Assign_2_B` and `Assign_2_C` containing the DrJava projects for the three parts of the assignment.

2. Using DrJava, print (to `CutePDF Writer`) the `.java` file of each of the parts for your assignment using the name *`ClassName`*`.pdf` where *`ClassName`* is the class name (i.e. same name as the `.java` file) and save the `.pdf` files at the **top level** of the assignment folder (i.e. directly within `Assign_2`).

3. Run the program for each part. When the display is finished (i.e. `Close` button visible), select `Print Image of Window...` from the `File` menu on the `TurtleDisplayer` and direct the output to `CutePDF Writer` and saving the `.pdf` file at the **top level** of the assignment folder (i.e. directly within `Assign_2`) using an appropriate name (e.g. `PartA.pdf`).
4. Create a `.zip` file of your submission by right-clicking on the top level folder (i.e. `Assign_2`) and selecting `Send to/Compressed (zipped) folder`. A zipped version of the folder will be created. Use the default name (`Assign_2.zip`).
5. Log on to Sakai and select the COSC 1P02 site.
6. On the `Assignments` page select `Assignment 2`. Attach your `.zip` file (e.g. `Assign_2.zip`) to the assignment submission (use the `Add/Remove Attachments` button and select `Browse`). Navigate to where you stored your assignment and select the `.zip` file (`Assign_2.zip`). The file will be added to your submission. Be sure to check the `Honor Pledge` checkbox. Press `Submit` to submit the assignment. You should receive a confirmation email.

DrJava

The `.zip` folder you submit should contain the project folders for the two parts, including all files relevant to the project—the `.java` and `.class` files for the assignment—and the `.pdf` files for program listings and output at the top level.

Other Platforms

If you are using an IDE other than DrJava to prepare your assignment, you must include the `.java` source files and the `.pdf` files described above for each part as well as an executable file (likely `.class` or`.jar`) that will execute on the lab machines.