CS2302 - Data Structures

Fall 2017

Lab 3

Due Friday, September 29, 2017

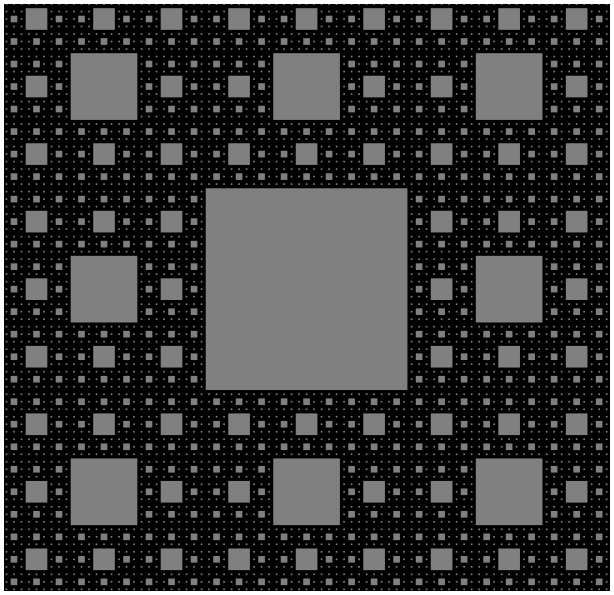
In this lab, you will practice using a stack as a substitute for recursion. Your task is to implement a non-recursive stack-based version of the following method:

```
public void drawGasket(int x, int y, int side) {
  int sub = side / 3;

  //Draw center square
  g2d.fill(new Rectangle2D.Double(x + sub, y + sub, sub - 1, sub - 1));

  if(sub >= 3) {
    //Draw 8 surrounding squares
    for (int i = 0; i < 3; i++){
      for (int j = 0; j < 3; j++){
        if (j!=1 || i != 1){
          drawGasket(x + i * sub, y + j * sub, sub);
        }
      }
    }
  }
}
```

The purpose of *drawGasket* is to draw what is called the *Sierpinski carpet**. When you call *drawGasket(0, 0, 729)*, you should see the following on your screen.



When implementing your non-recursive stack-based version of *drawGasket*, you must use a single stack. Feel free to use Java's generic Stack implementation. Since recursive calls are replaced with push and pop operations, each object in the stack must contain exactly the same parameters as a recursive call in the original version of the algorithm.

Use the following Java file as a starting point: https://www.dropbox.com/s/dzvvat3dewed8jg/SierpinskiLab.java

* https://en.wikipedia.org/wiki/Sierpinski_carpet

** The method *drawGasket* was adapted from Carl Burch's book *Programming via Java* (http://www.toves.org/books/java/ch18-recurex/).
As usual, write a report describing your results.