Recursive Programming has two key points:

1. Recursive function must call itself.
2. Recursive function must have a way to stop itself.

The recursiveDrawLine function is complex because each function call itself **TWICE**: one from start point to mid point, one from mid point to end point. This makes calls explosive in a rate 1 -> 2 -> 4 -> 8 -> 16.

The parameter recursiveNumber in recursiveDrawLine is used to stop function call. When call to itself, recursiveNumber is reduced by one. So sometime it will reach 0, and function will return in if(recursiveNumber==0) condition, which will not call itself again.

Below is a simple recursive program: print all even number. You can create a SimpleRecursion class in edu.nyu.cs.finale.drawLine package, copy the code to it, and run it. Look how recursive call stops. You can change UpLimit to any number.

**package** edu.nyu.cs.finale.drawLine;

/\*

\* Print even numbers to UpLimit (50)

\*/

**public** **class** SimpleRecursion {

**public** **static** **int** *UpLimit* = 50;

**public** **static** **void** main(String[] args) {

*printEvenNumber*(1);

}

**public** **static** **void** printEvenNumber(**int** i) {

**int** evenNumber = i\*2;

**if**(evenNumber > *UpLimit*) {

**return**;

}

System.***out***.println(evenNumber);

*printEvenNumber*(i+1);

}

}