REPORT

1.PROBLEM DESCRIPTION

In first problem, it is asked to make a mupliplication table that goes up to n\*n for the kids, but it must have symmetric positions as emptiness for the children to understand.If n=4, the table should appear like;

1

2 4

3 6 9

4 8 12 16

In the second problem, it is asked to write a series that finds the number of each previous number’s digits and muttiplies with n which is a constant number. After taking m for the first number, there is no previous number m is taken as the previous number. This must continue for 4 iterations.

2.PROBLEM SOLUTION

First,I used “public static final int” and “public static final double” for specifying the constants, N and M, that remains stable throughout the series. I used methods for each problem for seperating the problems and it’s codes.And also, I used “System.out.println();” for the seperate outputs of each problem.

Inside the first method, I used nested loop , outer loop for printing lines which determines the number of lines with its test section and the inner loop for printing patterns what the problem requires. And also, I used “System.out.println();” at the end of the inner loop to pass to the bottom line.

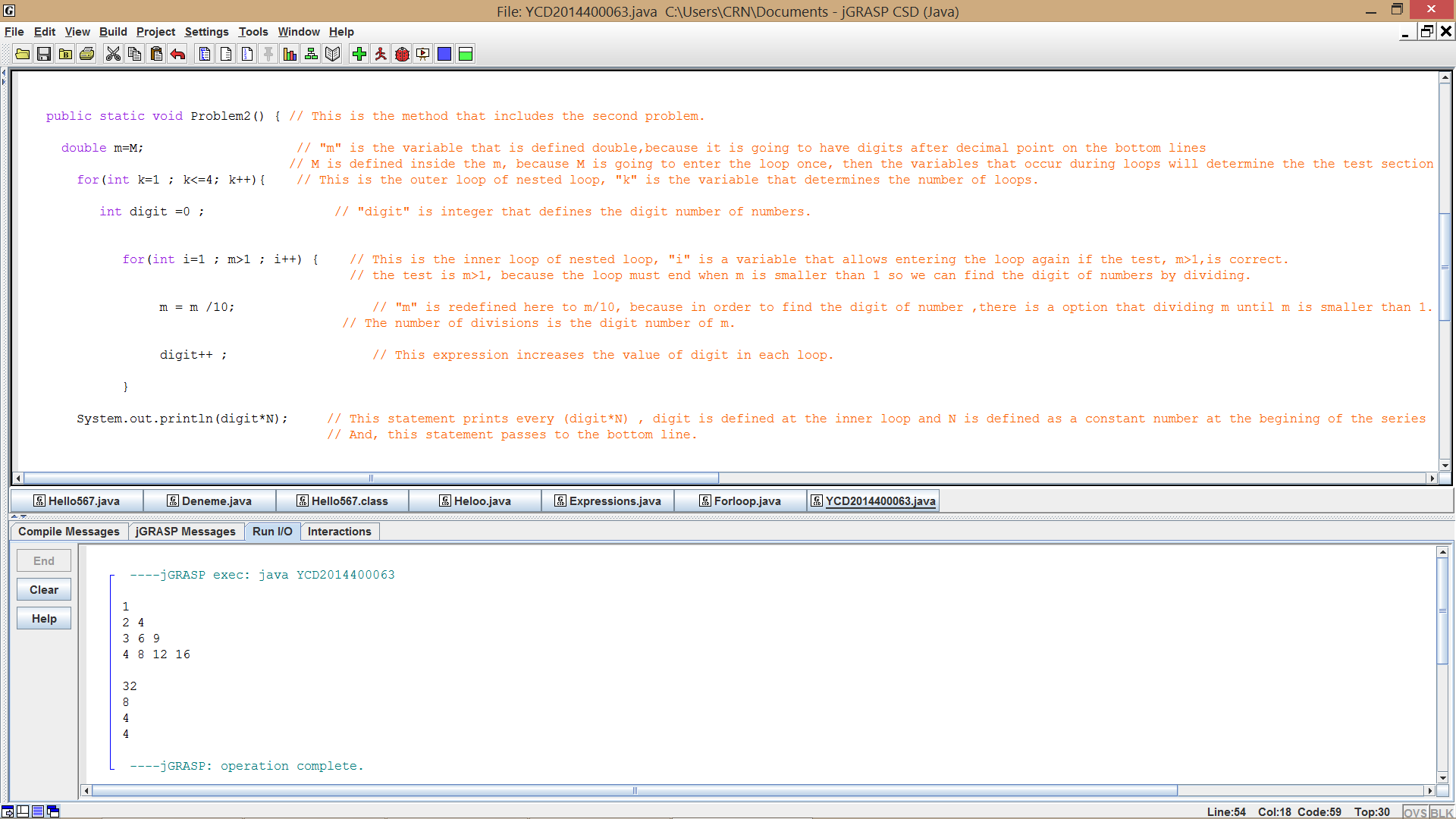
Inside the second method, I used double for specify the m=M to make M entered nested loop and provide what the problem asks, constant M must be taken only once. Then, I used nested loop, outer loop to specify the number of iterations and the inner loop to calculate the digit number of numbers. In order to find digit number, I used int and specify digit=0 and I used it in inner loop as digit++ to count the number of loops, because the number of loops shows the digit number of numbers that enters to the loop. In the inner loop, I use the calculation, m= m/10, and the test , m>1, because the loop will continue as long as the test fails; so we can calculate the digit number by specifying the test as m>1 and dividing the number to 10 until it is smaller than 1, the number of divisions will show the digit number of that number and it will automatically stop when m is smaller than 1. After the inner loop ends, I used “System.out.println(digit\*CONSTANT);” because the problem requires the multiplications of digit number and the constant for 4 iterations. Then, I redefined the m, because the problem wants that new m must be the multiplication of digit number and the constant number, not the previous one. This series will be continue for 4 iterations and will print the multiplication of digit number and the constant number.

3.IMPLEMENTATİTON

public class YCD2014400063 {  
  
 public static final int N=4; //N is the constant ,because in the second problem,there is a constant number that multiplies with digit at the end of each loop.And also,N is necessary for the first problem top rint n\*n.  
   
 public static final double M=12345678; //M is the constant for second problem, because there must be no previous number m is taken as the previous number.  
   
  
 public static void main(String[] args) {  
   
 Problem1(); //This statement is to call the method.  
 System.out.println(); // This statement is to seperate two outputs and to see the each problem's results clearly .  
 Problem2(); //This statement is to call the method.  
   
 }  
   
 public static void Problem1() { // This is the method that includes the first problem.  
 for(int line = 1 ; line <=N ; line++) { // This the outer loop of nested loop for specifying the number of lines." line" is the variable that specifies the number of lines.  
   
 for( int i =1; i <= line ; i++ ){ // This is the inner loop of nested loop to print the patterns of analysis. "i" is the variable to make the multiplication table as it is asked.  
   
 System.out.print (line\*i +" "); //This statement multiplies the line and i,prints ,puts a space and prints near the previous output again and again until test fails.  
   
 }  
 System.out.println(); // After the inner loop ends, this statement passes to the bottom line.  
   
 }  
   
   
 }

public static void Problem2() { // This is the method that includes the second problem.  
   
 double m=M; // "m" is the variable that is defined double,because it is going to have digits after decimal point on the bottom lines  
 // M is defined inside the m, because M is going to enter the loop once, then the variables that occur during loops will determine the the test section.   
 for(int k=1 ; k<=4; k++){ // This is the outer loop of nested loop, "k" is the variable that determines the number of loops.   
   
 int digit =0 ; // "digit" is integer that defines the digit number of numbers.  
   
   
 for(int i=1 ; m>1 ; i++) { // This is the inner loop of nested loop, "i" is a variable that allows entering the loop again if the test, m>1,is correct.  
 // the test is m>1, because the loop must end when m is smaller than 1 so we can find the digit of numbers by dividing.  
   
 m = m /10; // "m" is redefined here to m/10, because in order to find the digit of number ,there is a option that dividing m until m is smaller than 1.  
 // The number of divisions is the digit number of m.  
   
 digit++ ; // This expression increases the value of digit in each loop.  
   
 }  
   
 System.out.println(digit\*N); // This statement prints every (digit\*N) , digit is defined at the inner loop and N is defined as a constant number at the beginning of the series  
 // And, this statement passes to the bottom line.  
   
 m =digit\*N; // "m" is redefined here, the problem requires that new m must be the multiplication of the digit number of previous number and N ,the constant number.  
 }   
   
 }   
}

4.OUTPUT OF THE PROGRAM



5.CONCLUSION

I solved the problems, but I can write the second problem less complicated.