Extract Method

\*\*\*Assume all code is encapsulated into classes (obviously), I didn’t have the time to make that little change but I hope it still makes sense…

Take for instance this method that performs mathematical functions within a loop, prints the current values, and then returns the sum at the end of the iterations.

public double stupidPrintFunctions {

double x = Math.Random();

double y = Math.Random();

double z = Math.Random();

for (int i = 1; i < 1000; i++)

{

x = x + i;

y = y - i;

z = z\*i;

System.out.println("x is :" + x + ", y is :" + y + ", z is : " + z);

}

return x + y + z;

}

Somebody who is very bad at coding might think it's a good idea to extract the math functions within the loop into a method, as so:

public int stupidPrintFunction {

double x = Math.Random();

double y = Math.Random();

double z = Math.Random();

for (int i = 1; i < 1000; i++)

{

double n = performMathFunctions(x,y,z,i);

}

return n;

}

public double performMathFunctions(double x, double y, double z, int i)

{

double x = x + i;

double y = y - i;

double z = z \* i;

System.out.println("x is :" + x + ", y is :" + y + ", z is : " + z);

n = x + y + z;

}

Although at first glance one might think this code could work, it of course will not as we modify the x, y, and z variables stored within the stupidPrintFunction every iteration of the original for loop. However, in the refactored version, we never actually modify the variables stored in the stupidPrintFunction, but rather the temporary variables within the performMathFunctions() method. This means that subsequent calls to the performMathFunctions() method will not use the updated values for x, y, and z; only the i value of the call will change. This is because in Java, we can not return tuples, so we will never be able to update stupidPrintFunction’s x, y and z from the return of performMathFunctions.