**Source Code:**

package Spark

import com.sun.org.glassfish.external.statistics.AverageRangeStatistic

object Main2 extends App{

val lst = List((1, "alpha"), (2, "beta"), (3, "gamma"), (4, "zeta"), (5, "omega"))

println("Numbers where the corresponding string length is 4: ")

val result = lst.map{case (e1:Int, e2: String) => if (e2.length() ==4) println(e1)}

val rdd2 = for (x <- lst) yield if (x.\_2.length() == 4 ) x.\_1 else "0"

println("::::::::::::::")

println("Numbers for the corresponding string contains alphabet m or alphabet z: ")

val result2 = lst.map{case (e1:Int, e2: String) => if (e2.contains("m") || e2.contains("z")) println(e1)}

val rdd3 = for (x <- lst) yield if (x.\_2.contains("m") || x.\_2.contains("z")) x.\_1 else "0"

val rdd4 = rdd3.map(\_.toString.toInt) //To convert List[Any] to List[Int]

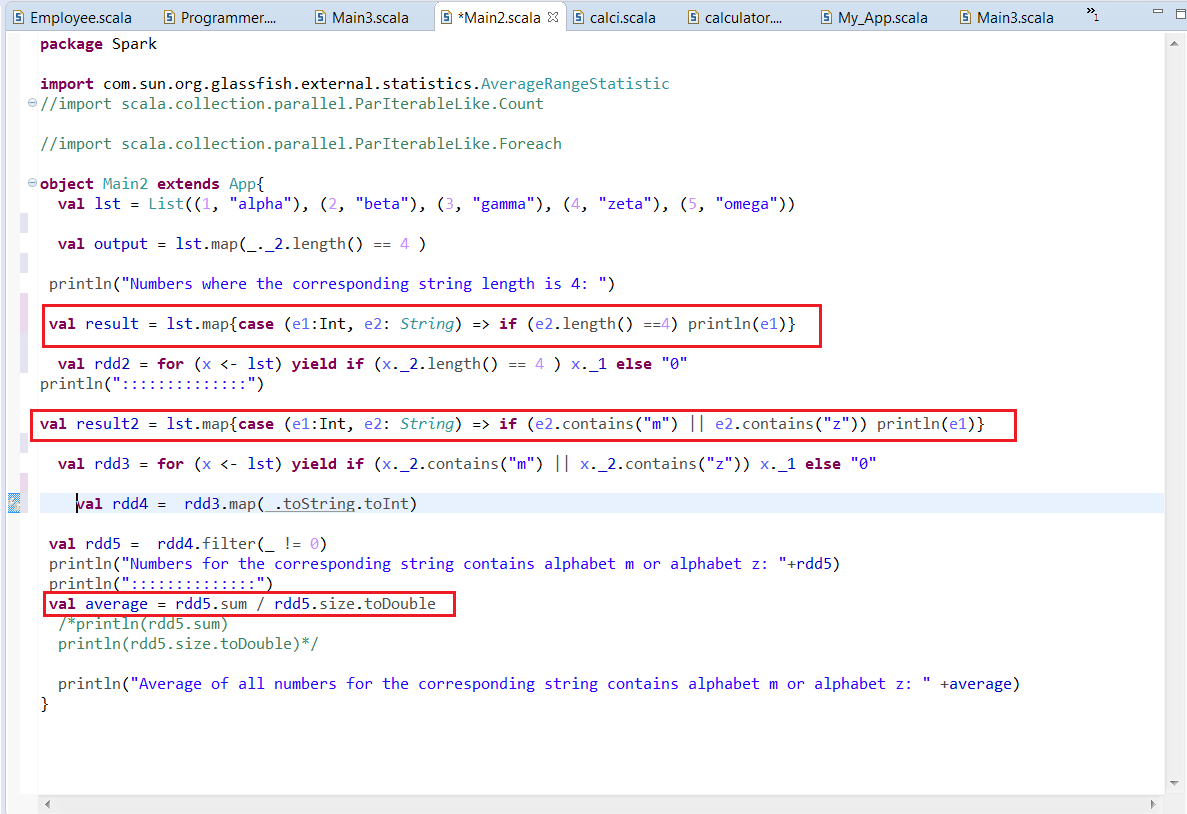
val rdd5 = rdd4.filter(\_ != 0)

println("::::::::::::::")

val average = rdd5.sum / rdd5.size.toDouble

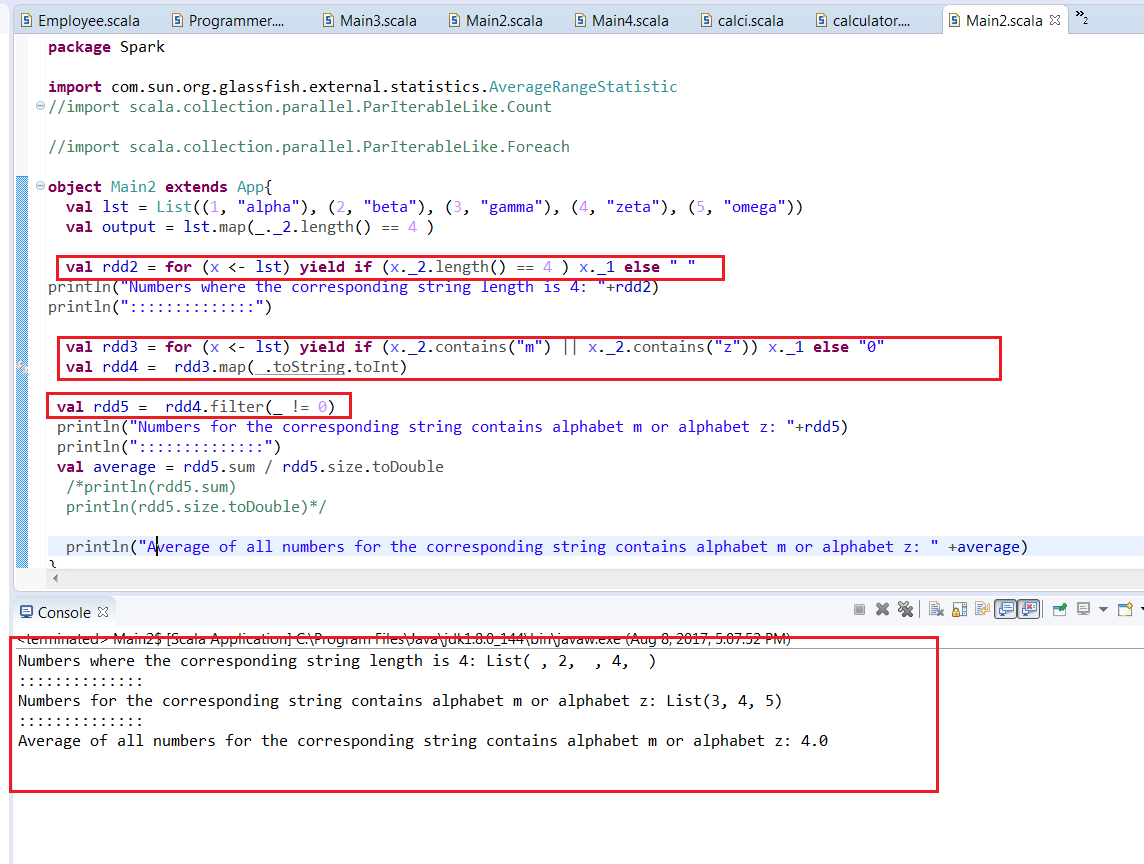
println("Average of all numbers for the corresponding string contains alphabet m or alphabet z: " +average)

}



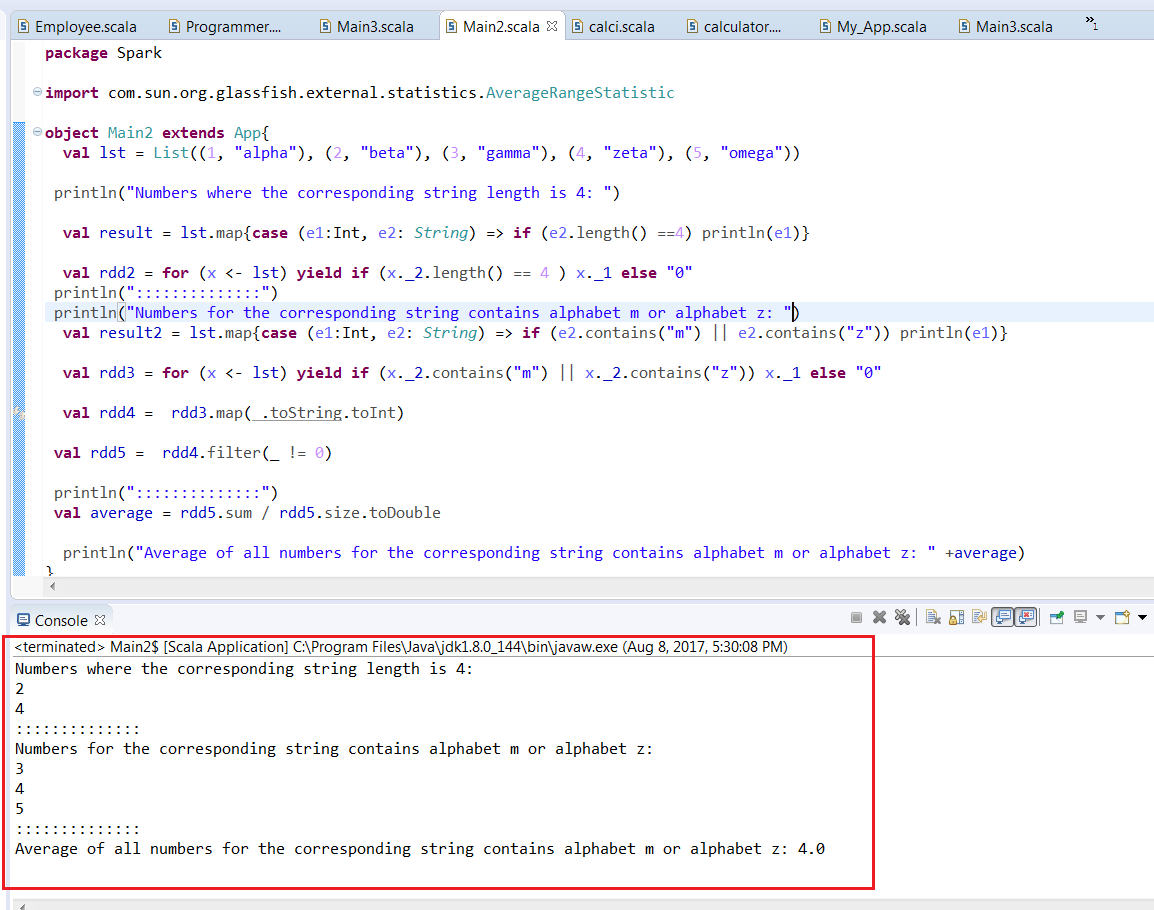
**Method –I :**

**Output in a List fashion:**



**Method –II:**

**Output in a number fashion:**



Note:

val result = lst.map{case (e1:Int, e2: String) => if (e2.length() ==4) println(e1)}

val result2 = lst.map{case (e1:Int, e2: String) => if (e2.contains("m") || e2.contains("z")) println(e1)}

These two commands will create output in a numbered fashion.