Session 13

Assignment 3 Questions

ACADGILD

Problem Statement

- ***Find square root of number using Babylonian method.***

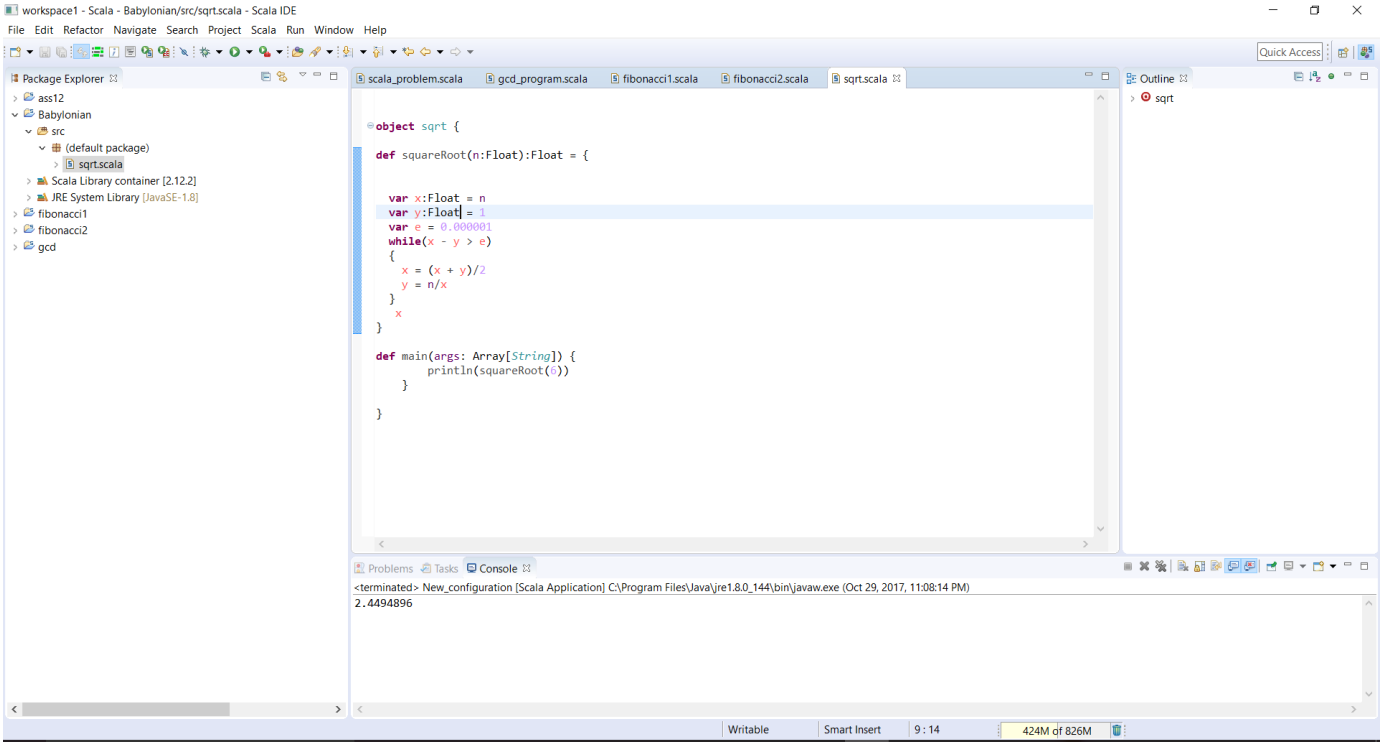
1 ***Start with an arbitrary positive start value x (the closer to the root, the better).***
2 ***Initialize y = 1.***
3 ***Do following until desired approximation is achieved.***
a) ***Get the next approximation for root using average of x and y***
b) ***Set y = n/x***

Program:-

```
object sqrt {

def squareRoot(n:Float):Float = {
```

```
  var x:Float = n
  var y:Float = 1
  var e = 0.000001
  while(x - y > e)
  {
    x = (x + y)/2
    y = n/x
  }
   x
}

def main(args: Array[String]) {
      println(squareRoot(6))
   }

}
```

Output: - 2.4494896


workspace1 - Scala - Babylonian/src/sqrt.scala - Scala IDE
File Edit Refactor Navigate Search Project Scala Run Window Help
Quick Access
Package Explorer
ass12
Babylonian
src
(default package)
sqrt.scala
Scala Library container [2.12.2]
JRE System Library [JavaSE-1.8]
fibonacci1
fibonacci2
gcd
scala_problem.scala
gcd_program.scala
fibonacci1.scala
fibonacci2.scala
sqrt.scala
object sqrt {
def squareRoot(n:Float):Float = {
var x:Float = n
var y:Float = 1
var e = 0.000001
while(x - y > e)
{
x = (x + y)/2
y = n/x
}
x
}
def main(args: Array[String]) {
println(squareRoot(6))
}
}
Outline
sqrt
Problems Tasks Console
<terminated> New_configuration [Scala Application] C:\Program Files\Java\jre1.8.0_144\bin\javaw.exe (Oct 29, 2017, 11:08:14 PM)
2.4494896
Writable
Smart Insert
9 : 14
424M of 826M