Session 16:
SCALA BASICS 3

Assignment 1

Task 1
Create a calculator to work with rational numbers.
Requirements:
➢ It should provide capability to add, subtract, divide and multiply rational Numbers
➢ Create a method to compute GCD (this will come in handy during operations on rational)
Add option to work with whole numbers which are also rational numbers i.e. (n/1)
➢ achieve the above using auxiliary constructors
➢ enable method overloading to enable each function to work with numbers and rational.

Solution :

Defined a Class

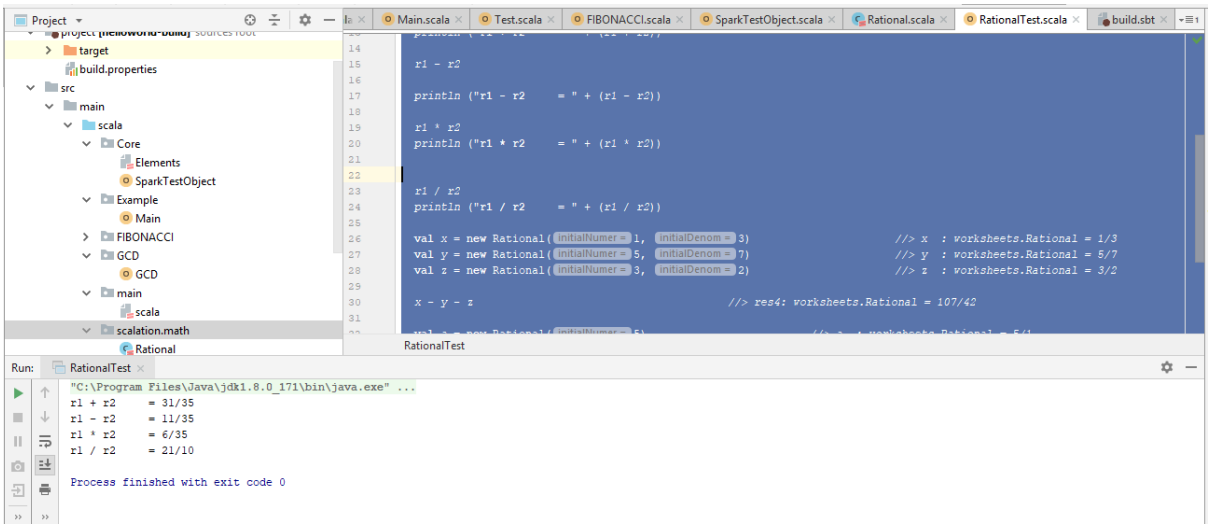
```
package Rational.math

class Rational(initialNumer: Int = 1, initialDenom: Int = 1) {
  require(initialDenom != 0, "denominator must be nonzero")

  private val gcd = {
    def gcdRec(x: Int, y: Int): Int = {
      if(y == 0) x else gcdRec(y, x % y)
    }
    gcdRec(initialNumer, initialDenom)
  }

  val numer = (if(initialDenom < 0) -(initialNumer) else (initialNumer)) / gcd
  val denom = (initialDenom) / gcd

  def unary_- = new Rational(-numer, denom)

  def +(that: Rational) =
    new Rational(that.numer * denom + numer * that.denom,
      denom * that.denom)

  def +(that: Int): Rational =
```

```
    this + (new Rational(that, 1))

  def -(that: Rational) = this + -that
  def -(that: Int): Rational = this - new Rational(that)

  def *(that: Rational) = new Rational(numer * that.numer, denom * that.denom)
  def *(that: Int): Rational = this * new Rational(that)

  def /(that: Rational) = new Rational(numer * that.denom, denom * that.numer)
  def /(that: Int): Rational = this / new Rational(that)


  override def toString = numer + "/" + denom
}
```

Defined a Object:

```
package Rational.math

object RationalTest extends App {


  val r1 = new Rational(3, 5)
  val r2 = new Rational(2, 7)
  val r3 = new Rational(5, 9)

  r1 + r2

  println ("r1 + r2     = " + (r1 + r2))

  r1 - r2

  println ("r1 - r2     = " + (r1 - r2))

  r1 * r2
  println ("r1 * r2     = " + (r1 * r2))


  r1 / r2
  println ("r1 / r2     = " + (r1 / r2))

  }
```

Output:


Project
target
build.properties
src
main
scala
Core
Elements
SparkTestObject
Example
Main
FIBONACCI
GCD
GCD
main
scala
scalation.math
Rational
Main.scala
Test.scala
FIBONACCI.scala
SparkTestObject.scala
Rational.scala
RationalTest.scala
build.sbt
r1 - r2
println ("r1 - r2 = " + (r1 - r2))
r1 * r2
println ("r1 * r2 = " + (r1 * r2))
r1 / r2
println ("r1 / r2 = " + (r1 / r2))
val x = new Rational(initialNumer = 1, initialDenom = 3) //> x : worksheets.Rational = 1/3
val y = new Rational(initialNumer = 5, initialDenom = 7) //> y : worksheets.Rational = 5/7
val z = new Rational(initialNumer = 3, initialDenom = 2) //> z : worksheets.Rational = 3/2
x - y - z //> res4: worksheets.Rational = 107/42
RationalTest
Run:
RationalTest
"C:\Program Files\Java\jdk1.8.0_171\bin\java.exe" ...
r1 + r2 = 31/35
r1 - r2 = 11/35
r1 * r2 = 6/35
r1 / r2 = 21/10
Process finished with exit code 0