BIG DATA HADOOP & SPARK TRAINING

Assignment on Scala III

Rashmi Krishna 4/23/18

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

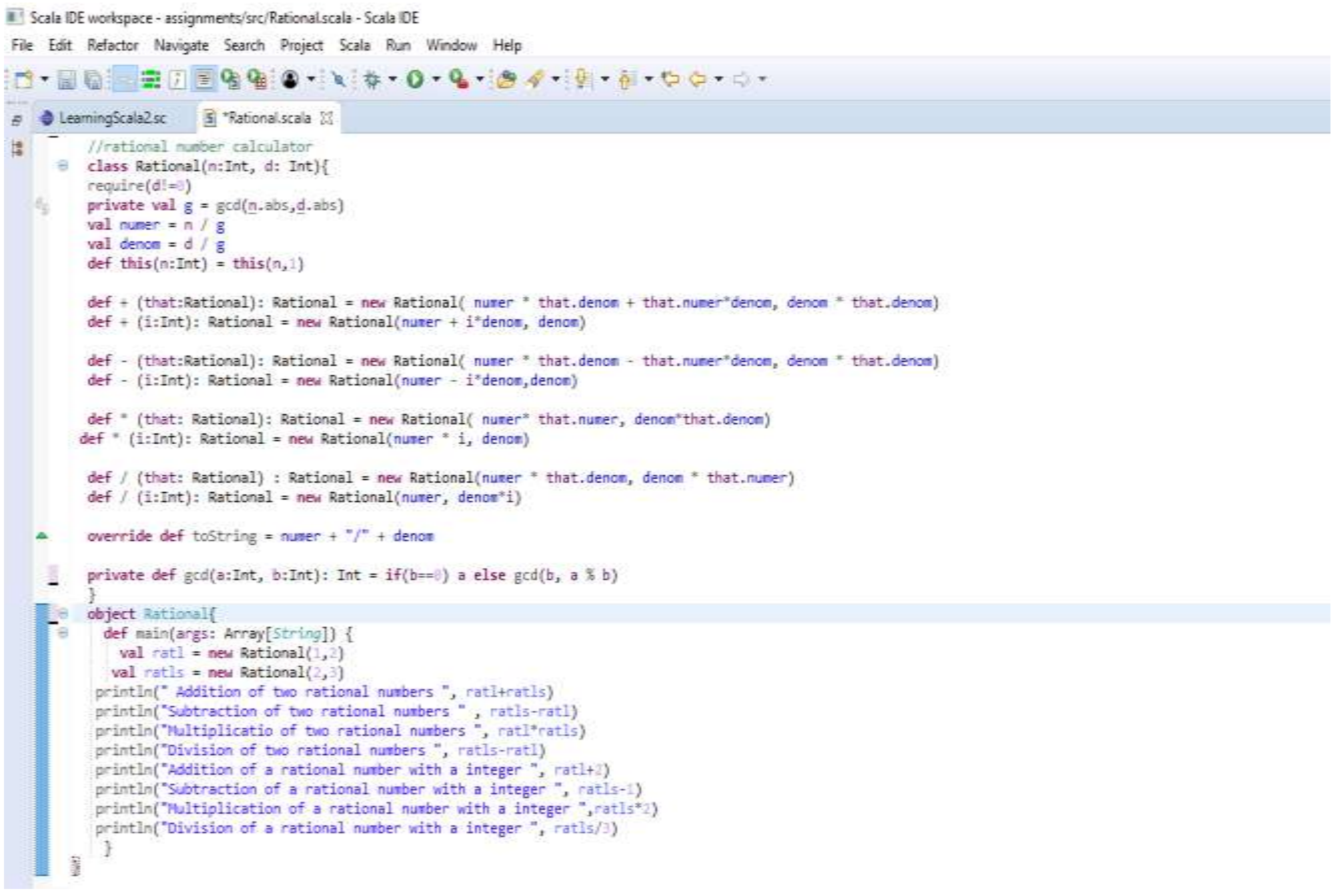
➢ enable method overloading to enable each function to work with numbers and rational.

```
 //rational number calculator
 class Rational(n:Int, d: Int){// initializing the parameterized class Rational with two
parameters n and d
 require(d!=0)// this indicates that denominator should not be zero
 private val g = gcd(n.abs,d.abs)
 val numer = n / g
 val denom = d / g
 def this(n:Int) = this(n,1)
```

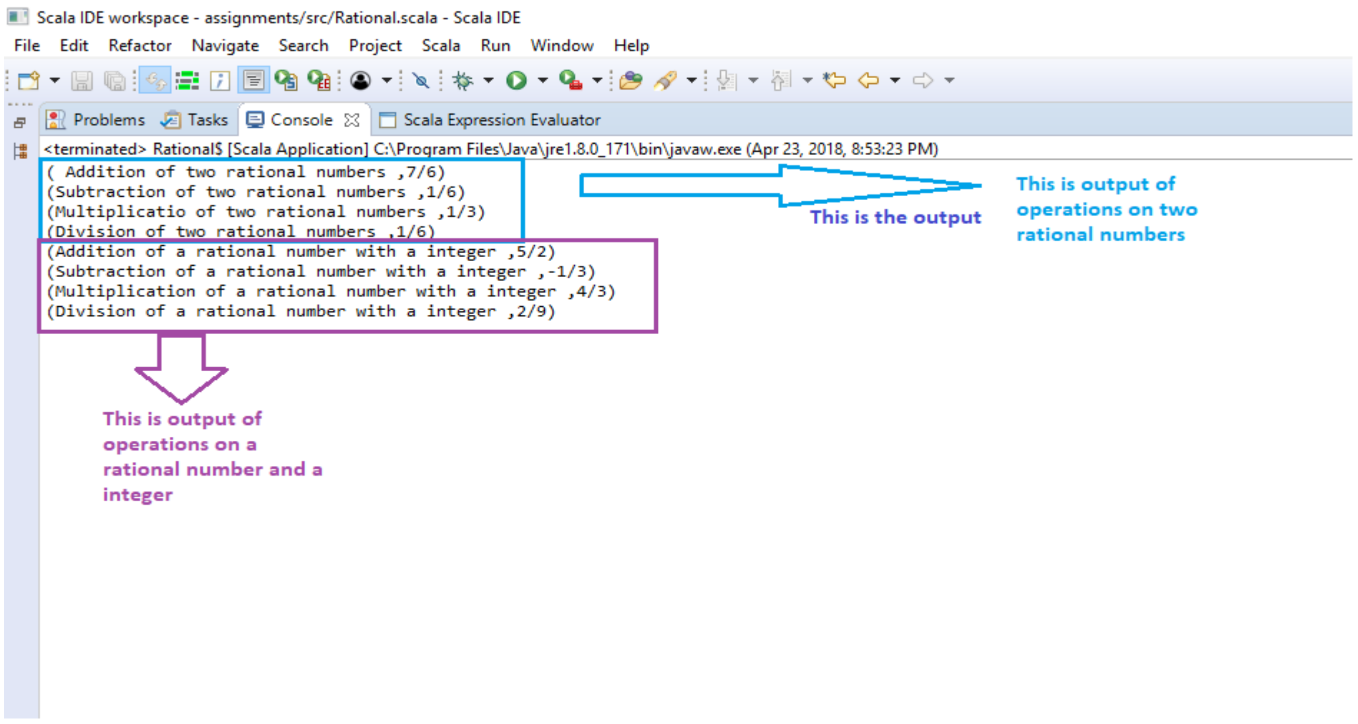
for eg: if input rational number is 12/18 then it must be divided by highest multiple, as in this example here highest multiple is 6, so the actual rational number is 2/3. To get this we are using the gcd function

```
// To work with whole numbers which are also rational numbers we are using auxiliary
constructor

 def + (that:Rational): Rational = new Rational( numer * that.denom + that.numer*denom,
denom * that.denom)//function to add two rational numbers
 def + (i:Int): Rational = new Rational(numer + i*denom, denom)
 // function to add a rational number with a integer( we are overloading + function)
 def - (that:Rational): Rational = new Rational( numer * that.denom - that.numer*denom,
denom * that.denom) //function to subtract two rational numbers
 def - (i:Int): Rational = new Rational(numer - i*denom,denom)
   // function to subtract a rational number with a integer( we are overloading - function)
 def * (that: Rational): Rational = new Rational( numer* that.numer, denom*that.denom)
//function to multiply two rational numbers
def * (i:Int): Rational = new Rational(numer * i, denom)
   // function to multiply a rational number with a integer( we are overloading * function)

 def / (that: Rational) : Rational = new Rational(numer * that.denom, denom * that.numer)
//function to divide two rational numbers
 def / (i:Int): Rational = new Rational(numer, denom*i)
   // function to divide a rational number with a integer ( we are overloading / function)
 override def toString = numer + "/" + denom // to display the output in the format n/d
we have to override as we are overloading functions using auxiliary constructor

 private def gcd(a:Int, b:Int): Int = if(b==0) a else gcd(b, a % b)
 }
 object Rational{
  def main(args: Array[String]) {
    val ratl = new Rational(1,2)
   val ratls = new Rational(2,3)
  println(" Addition of two rational numbers ", ratl+ratls)
  println("Subtraction of two rational numbers " , ratls-ratl)
  println("Multiplicatio of two rational numbers ", ratl*ratls)
  println("Division of two rational numbers ", ratls-ratl)
  println("Addition of a rational number with a integer ", ratl+2)
  println("Subtraction of a rational number with a integer ", ratls-1)
  println("Multiplication of a rational number with a integer ",ratls*2)
  println("Division of a rational number with a integer ", ratls/3)
   }
}
```

```
// instantiating Rational class by passing two
numbers as arguments
```


Scala IDE workspace - assignments/src/Rational.scala - Scala IDE
File Edit Refactor Navigate Search Project Scala Run Window Help
LearningScala2.sc
*Rational.scala

```
//rational number calculator
class Rational(n:Int, d: Int){
require(d!=0)
private val g = gcd(n.abs,d.abs)
val numer = n / g
val denom = d / g
def this(n:Int) = this(n,1)

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

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

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

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

override def toString = numer + "/" + denom

private def gcd(a:Int, b:Int): Int = if(b==0) a else gcd(b, a % b)
}
object Rational{
  def main(args: Array[String]) {
    val ratl = new Rational(1,2)
   val ratls = new Rational(2,3)
 println(" Addition of two rational numbers ", ratl+ratls)
 println("Subtraction of two rational numbers " , ratls-ratl)
 println("Multiplicatio of two rational numbers ", ratl*ratls)
 println("Division of two rational numbers ", ratls-ratl)
 println("Addition of a rational number with a integer ", ratl+2)
 println("Subtraction of a rational number with a integer ", ratls-1)
 println("Multiplication of a rational number with a integer ",ratls*2)
 println("Division of a rational number with a integer ", ratls/3)
  }
```


Scala IDE workspace - assignments/src/Rational.scala - Scala IDE
File Edit Refactor Navigate Search Project Scala Run Window Help
Problems Tasks Console Scala Expression Evaluator
This is the output
This is output of operations on two rational numbers
This is output of operations on a rational number and a integer

```
<terminated> Rational$ [Scala Application] C:\Program Files\Java\jre1.8.0_171\bin\javaw.exe (Apr 23, 2018, 8:53:23 PM)
( Addition of two rational numbers ,7/6)
(Subtraction of two rational numbers ,1/6)
(Multiplicatio of two rational numbers ,1/3)
(Division of two rational numbers ,1/6)
(Addition of a rational number with a integer ,5/2)
(Subtraction of a rational number with a integer ,-1/3)
(Multiplication of a rational number with a integer ,4/3)
(Division of a rational number with a integer ,2/9)
```