

Assignment - 16.1

Task1: 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)

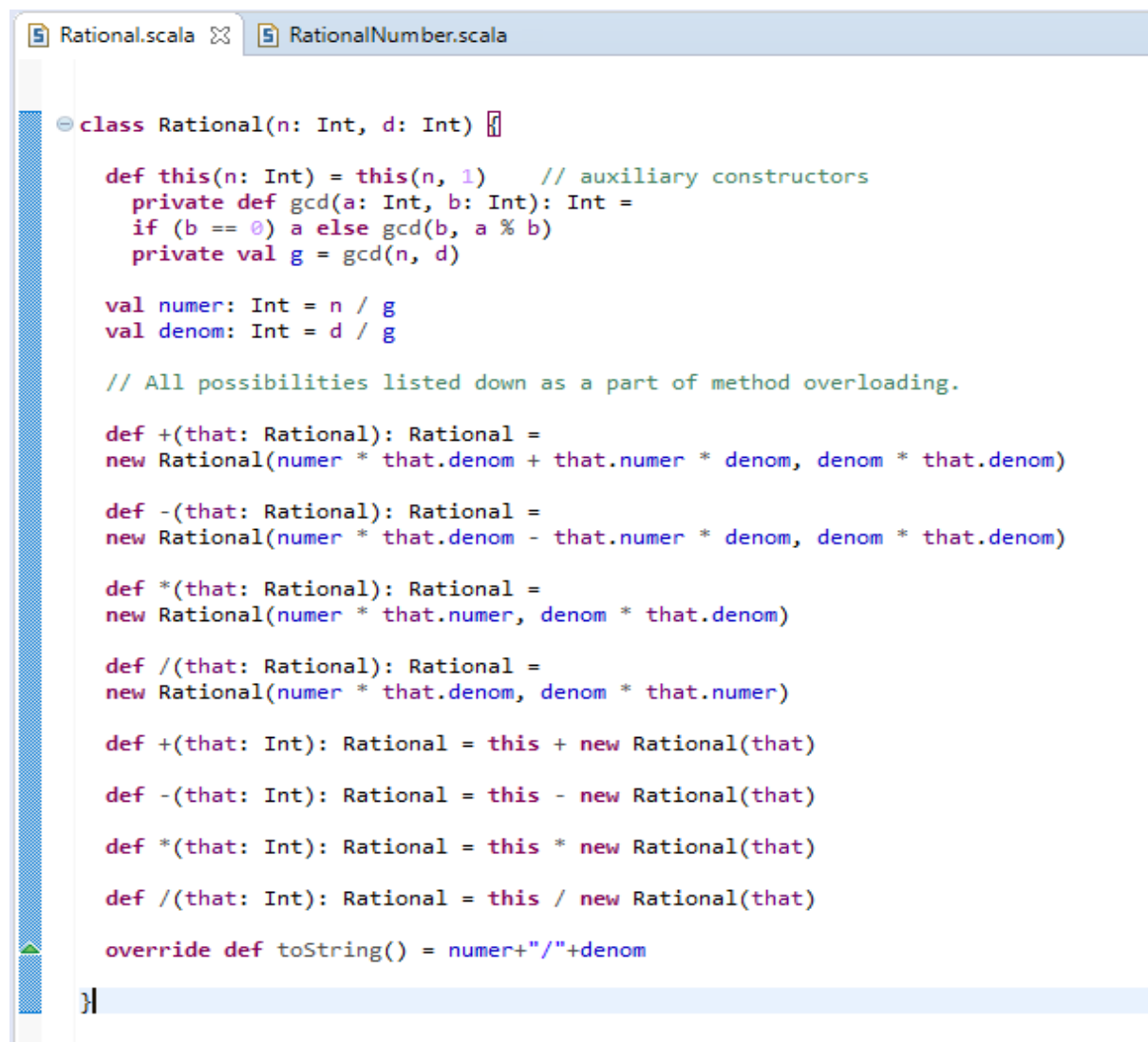
Add option to work with whole numbers which are also rational numbers i.e. (n/1)

- Achieve the above using auxiliary constructors
- Enable method overloading to enable each function to work with numbers and rational.

Answer:

Here we are creating Scala Class as well as Scala Object to execute the same. The code is mentioned in below screenshot.

Rational.scala



```
class Rational(n: Int, d: Int) {  
  def this(n: Int) = this(n, 1) // auxiliary constructors  
  private def gcd(a: Int, b: Int): Int =  
    if (b == 0) a else gcd(b, a % b)  
  private val g = gcd(n, d)  
  
  val numer: Int = n / g  
  val denom: Int = d / g  
  
  // All possibilities listed down as a part of method overloading.  
  
  def +(that: Rational): Rational =  
    new Rational(numer * that.denom + that.numer * denom, denom * that.denom)  
  
  def -(that: Rational): Rational =  
    new Rational(numer * that.denom - that.numer * denom, denom * that.denom)  
  
  def *(that: Rational): Rational =  
    new Rational(numer * that.numer, denom * that.denom)  
  
  def /(that: Rational): Rational =  
    new Rational(numer * that.denom, denom * that.numer)  
  
  def +(that: Int): Rational = this + new Rational(that)  
  def -(that: Int): Rational = this - new Rational(that)  
  def *(that: Int): Rational = this * new Rational(that)  
  def /(that: Int): Rational = this / new Rational(that)  
  
  override def toString() = numer+"/"+denom  
}
```

RationalNumber.scala

```
Rational.scala RationalNumber.scala
```

```
object RationalNumber {  
  def main(args: Array[String]): Unit = {  
    println("Hello, world!")  
  
    val x = new Rational(2, 3)  
    val y = new Rational(3, 4)  
  
    val a = x * x  
    println("Result a: " + a)  
  
    val b = a * 2  
    println("Result b: " + b)  
  
    val z = (x + y) * x  
    println("Result z: " + z)  
  
    implicit def intToRational(x: Int) = new Rational(x)  
    val r = new Rational(2, 3)  
    val s = 2 * r  
    println("Result s: " + s)  
  }  
}
```

Once code is ready, execute it by using “Run As ➔ Scala Application”. The output will be as received.

```
Problems Tasks Console Git Staging
```

```
<terminated> RationalNumber$ [Scala Application] C:\Program Files\Java\jre1.8.0_171\bin\javaw.exe (10-May-2018, 12:38:35 AM)  
Hello, world!  
Result a: 4/9  
Result b: 8/9  
Result z: 17/18  
Result s: 4/3
```