Assignment 16: Scala Basics 3 Assignment Problems

Problem Statement

==========

Task 1

Create a calculator to work with rational numbers.
Requirements:
➢ It should provide capability to add, subtract, divide and multiply rational Numbers

Terminal Execution :

```
scala> class Calculator{ private var result = 0
     | def add(a:Int,b:Int): Unit = result = a + b
     | def subtract(a:Int,b:Int):Unit = if(a > b) result = (a - b)  else result = (b - a)
     | def divide(a:Int,b:Int):Unit = if(b != 0) result = (a / b)  else result = 0
     | def multiply(a:Int,b:Int):Unit = result = a * b
     | def printResult():Int = result
     | }
defined class Calculator


scala> var calresult = new Calculator
calresult: Calculator = Calculator@124411f

scala> calresult.add(2,4)
scala> calresult.printResult()
res1: Int = 6

scala> calresult.subtract(5,10)
scala> calresult.printResult()
res3: Int = 5

scala> calresult.divide(10,5)
scala> calresult.printResult()
res5: Int = 2

scala> calresult.multiply(10,6)
scala> calresult.printResult()
```

res7: Int = 60

==========

➢ Create a method to compute GCD (this will come in handy during operations on rational)

Terminal Execution :

```
scala> class Calculator{ private var result = 0
     | def add(a:Int,b:Int): Unit = result = a + b
     | def subtract(a:Int,b:Int):Unit = if(a > b) result = (a - b)  else result = (b - a)
     | def divide(a:Int,b:Int):Unit = if(b != 0) result = (a / b)  else result = 0
     | def multiply(a:Int,b:Int):Unit = result = a * b
     | def printResult():Int = result
     | def gcd(a:Int,b:Int):Int = { if(b==0) a  else gcd(b,a%b)}
     | def gcdResult(a:Int, b:Int):Unit = result = gcd(a,b)
     | }
defined class Calculator

scala> var cal = new Calculator
cal: Calculator = Calculator@1b266b3

scala> cal.gcdResult(20,50)

scala> cal.printResult()
res9: Int = 10
```

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

➢ **Achieve the above using auxiliary constructors**

Terminal Execution :

```
scala> class Calculator(a:Int,b:Int,c:Int,d:Int) { private var result = 0
     | val num1 = a
     | val deno1 = b
     | val num2 = c
     | val deno2 = d
     | def this(a:Int,c:Int) = this(a,2,c,4)
     | def add(): Unit = {
     | if(deno1 != 0 && deno2 != 0) result = ((num1 / deno1) + (num2/deno2)) else result = 0
     | }
     | def subtract(): Unit = {
     | if(deno1 != 0 && deno2 != 0) result = ((num1 / deno1) - (num2/deno2)) else result = 0
     | }
     | def multiply(): Unit = {
     | if(deno1 != 0 && deno2 != 0) result = ((num1 / deno1) * (num2/deno2)) else result = 0
     | }
     | def divide(): Unit = {
     | if(deno1 != 0 && num2 != 0) result = ((num1 / deno1) / (num2/deno2)) else result = 0
     | }
     | def printResult():Int = result
     | }
defined class Calculator

scala> var cal = new Calculator(6,4,2,3)
cal: Calculator = Calculator@6ba0ac

scala> cal.add()

scala> cal.printResult()
res2: Int = 1

scala> cal.subtract()

scala> cal.printResult()
res4: Int = 1
```

```
scala> cal.multiply()

scala> cal.printResult()
res6: Int = 0

scala> var cal = new Calculator(10,16)
cal: Calculator = Calculator@1e18270

scala> cal.add()

scala> cal.printResult()
res8: Int = 9

scala> cal.subtract()

scala> cal.printResult()
res10: Int = 1

scala> cal.multiply()

scala> cal.printResult()
res12: Int = 20

scala> cal.divide()

scala> cal.printResult()
res14: Int = 1
```

==========

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

Terminal Execution :

```
scala> class Calculator(a:Int,b:Int,c:Int,d:Int) { private var result = 0
     | val num1 = a
     | val deno1 = b
     | val num2 = c
     | val deno2 = d
     | def this(a:Int,c:Int) = this(a,2,c,4)
     | def add(): Unit = {
```

```
     | if(deno1 != 0 && deno2 != 0) result = ((num1 / deno1) + (num2/deno2)) else result = 0
     | }
     | def add(a:Int,c:Int): Unit = result = a + c
     | def subtract(): Unit = {
     | if(deno1 != 0 && deno2 != 0) result = ((num1 / deno1) - (num2/deno2)) else result = 0
     | }
     | def multiply(): Unit = {
     | if(deno1 != 0 && deno2 != 0) result = ((num1 / deno1) * (num2/deno2)) else result = 0
     | }
     | def multiply(a:Int,c:Int): Unit = result = a * c
     | def divide(): Unit = {
     | if(deno1 != 0 && num2 != 0) result = ((num1 / deno1) / (num2/deno2)) else result = 0
     | }
     | def divide(a:Int,c:Int): Unit = {
     | if(c != 0) result = a/c else result = 0
     | }
     | def printResult():Int = result
     | }
defined class Calculator

scala> var cal = new Calculator(6,4,2,3)
cal: Calculator = Calculator@adbc9d

scala> cal.add()

scala> cal.printResult()
res16: Int = 1

scala> cal.add(4,8)

scala> cal.printResult()
res18: Int = 12

scala> var cal = new Calculator(6,4)
cal: Calculator = Calculator@1f80c04

scala> cal.subtract()

scala> cal.printResult()
res20: Int = 2

scala> cal.multiply()

scala> cal.printResult()
res23: Int = 3
```

```
scala> cal.multiply(4,6)

scala> cal.printResult()
res25: Int = 24

scala> cal.multiply()

scala> cal.printResult()
res23: Int = 3

scala> cal.multiply(4,6)

scala> cal.printResult()
res25: Int = 24

scala> cal.divide()

scala> cal.printResult()
res27: Int = 3

scala> cal.divide(10,2)

scala> cal.printResult()
res29: Int = 5

scala> cal.divide(10,0)

scala> cal.printResult()
res31: Int = 0
```