

Scala in Practice

lab 02

Acceptance criteria:

Create Scala program with:

- Package *numbers* which contains class representing rational numbers:
 - **class** Rational(. . .) {


```

          ...
          def +(other: Rational): Rational = ??? // addition
          def -(other: Rational): Rational = ??? // subtraction
          def *(other: Rational): Rational = ??? // multiplication
          def /(other: Rational): Rational = ??? // division
          }
```
 - Validate arguments of constructors with *require*¹ precondition
 - *toString* should return the reduced fraction (for example: Rational(50, 6).*toString* => "8 1/3")
 - Create the *companion object* with factory methods for numbers: 0 & 1 & *apply* with default denominator = 1
- Package *figures* which contains:
 - Classes to model *triangle*, *rectangle* & *square* in Euclidean space, all defined on vertices being instances of **class** Point(x: Rational, y: Rational). Having functions:
 - **def** area: Double = ???
 - **val** description: String = ??? //"Triangle", "Rectangle" or "Square"
 - Propose couple *auxiliary constructors* or *companion objects* with functions which could ease usage of these classes
- Create *singleton* object with functions:
 - **def** areaSum(figures: List[...]): Double = ??? //Sum all areas
 - **def** printAll(figures: List[...]): Unit = ??? //Print all descriptions
- Create *application entry-point* object with some example tests for the above implementation

Michał Kowalczykiewicz

¹ [https://www.scala-lang.org/api/current/scala/Predef\\$.html](https://www.scala-lang.org/api/current/scala/Predef$.html)