FICHA FORMATIVA

Consider the project for this session available at https://bitbucket.org/upskills-isep/simplealgorithmsclass/src/master/ . It uses a type alias, which can facilitate understanding by simplifying complex types. However, aliases can be used even for simple Scala types.

Analyse the project, including the tests.

Some resources can be useful, namely:

- https://en.wikipedia.org/wiki/Rectangle
- https://www.cuemath.com/geometry/isosceles-right-triangle/

1.EXERCISES: ALGORITHMS RELATED TO RECTANGLES

1. Write the functions whose templates are provided and complement their tests:
    a. Write the function `perimeter`.
    b. Write the function `area`.
    c. Write the function `r2String`.
    d. Write the function `fPerimeter`.
    e. Write the function `fArea`.
    f. Write the function `sortRectangles`.
    g. Write a function `greatestTriangle` that receives a representation of Rectangle and returns the dimension of the two equal sides of the largest Isosceles Right Triangle that allows the rectangle to be completely filled. For example, a rectangle with sides 8 and 12 in a given unit of measure can be completely filled with an Isosceles Right Triangle with two sides of dimension 4 (in the same unit of measure) and no larger one would allow it.
    h. Write a function `numberOfSquares` that receives a representation of Rectangle and an integer value `d` and returns how many squares with dimension `d` allow the rectangle to be completely or nearly completely filled.
    i. Write a function `sumsidesR` that receives a list of representations of Rectangle and returns a tuple with the sum of their sides, without using loops or recursion, but the `reduceLeft` operation.

j. Write a function `sumsidesJ` that receives a list of representations of Rectangle and returns a tuple with the sum of their sides, the sum of the shortest side must be the first element of the tuple. Do not use loops or recursion, but the `joinLeft` operation.

k. Write a function `mapByF` that receives a list of representations of Rectangle and a function and returns a map where the key is the value of the function as indicated in the test(s). Do not use loops or recursion.

l. Write a function `mapWithUniqueByF`, similar to `mapByF`, that uses a Set to avoid repetition of rectangles. Do not use loops or recursion.